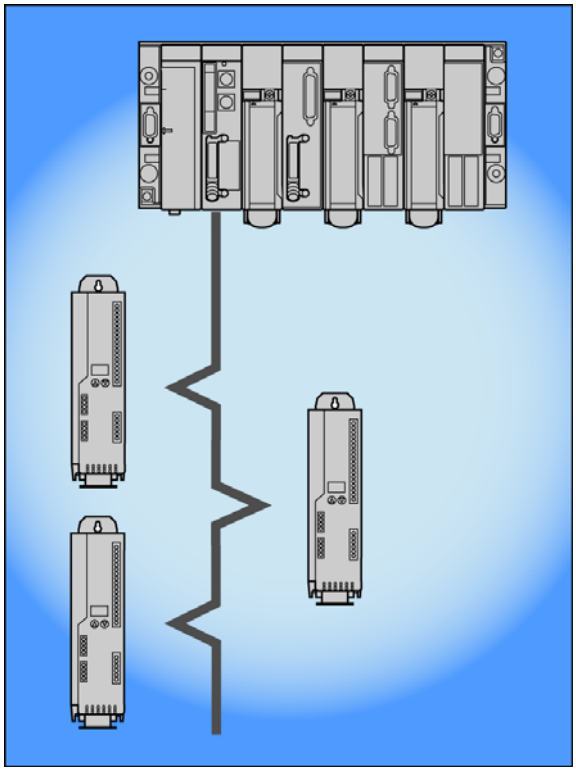
# Lexium15

## CANopen

User's manual

Retain for future use



a brand of Schneider Electric

Telemecanique

# Contents


Contents 3

Important information 4

Before you begin 5

CANOpen Wiring 6
- Using the AM02CAN001V000 accessory 6
- Without the AM02CAN001V000 accessory 6
- Bus cable specifications 7

CANopen on Lexium 15 8
- Standards 8

CANopen overview 9
- PLCopen Motion Function Blocks for Schneider Motion Controllers 9
- PDO and SDO notions 10
- PDO mapping configuration 11
- Lexium 15 predefined PDO 12
- SDO handling 13

Device overview 14

Device control 15
- Device Control Graph for Lexium 15 LP 15
- Device Control Graph for Lexium 15 MP/HP 17
- Control Word 19
- Modes of operation 20
- Status Word 21

Servo drive functions 22
- Digital inputs operations 22
- Digital Outputs operations 23
- Analog input operations 24
- Analog output operations 25
- Registration (LATCH) 25
- Position register 26
- Modes of operation 28
- Position profile (Point to point) (DSP 402) 31
- Velocity profile (DSP 402) 33
- Profile torque (DSP 402) 34
- Position Controller Data 34
- Position Control (DSP 402) 35
- Jog (only for Lexium 15 MP/HP) 35
- Digital Velocity (only for Lexium 15 MP/HP) 36
- Digital torque (only for Lexium 15 MP/HP) 36
- Gear manufacturer mode 37
- Motion Task 38
- Manufacturer Data (Manufacturer status data) 40
- Factor group 41
- Identification 46
- Errors 47
- Warnings/ Manufacturer Status Register 49

CANOpen profile 50
- Lexium 15 MP/HP 50
- Basic data types 51
- Communication Objects 52
- Abort SDO Protocol 54
- Heartbeat (only for Lexium 15 LP) 57

# Important information

## NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personnal injury if the instruction are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

**⚠ DANGER**

DANGER indicates an imminently hazardous situation, which, if not avoided, **will result** in death, serious injury, or equipment damage.

**⚠ WARNING**

Warning indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

**⚠ CAUTION**

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

PLEASE NOTE

Electrical equipment should be serviced only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material. This document is not intended as an instruction manual for untrained persons.

# Before you begin

**Read and understand these instructions before performing any procedure with this servo drive.**

## ⚠ DANGER

**HAZARDOUS VOLTAGE**

- Read and understand this manual before installing or operating the Lexium 15 servo drive. Installation, adjustment, repair, and maintenance must be performed by qualified personnel.
- The user is responsible for compliance with all international and national electrical standards in force concerning protective grounding of all equipment.
- Many parts in this variable speed servo drive, including printed wiring boards, operate at line voltage. DO NOT TOUCH. Use only electrically insulated tools.
- DO NOT touch unshielded components or terminal strip screw connections with voltage present.
- DO NOT short across terminals PA and PC or across the DC bus capacitors.
- Install and close all covers before applying power or starting and stopping the servo drive.
- Before servicing the variable speed servo drive:
  - Disconnect all power
  - Place a "DO NOT TURN ON" label on the variable speed servo drive disconnect
  - Lock the disconnect in the open position

-Disconnect all power including the external control power supply that may be present before servicing the servo drive. WAIT 15 MINUTES for the DC bus capacitors to discharge. Then follow the DC bus voltage measurement procedure given in the Installation Manual to verify that the DC voltage is less than ⎓ 45 V. The servo drive LEDs are not accurate indicators of the absence of DC bus voltage.

**Failure to follow these instructions will result in death, serious injury, or equipment damage.**

## ⚠ CAUTION

**DAMAGED EQUIPMENT**

Do not operate or install any servo drive that appears damaged.

**Failure to follow this instruction can result in injury or equipment damage.**

# CANOpen Wiring

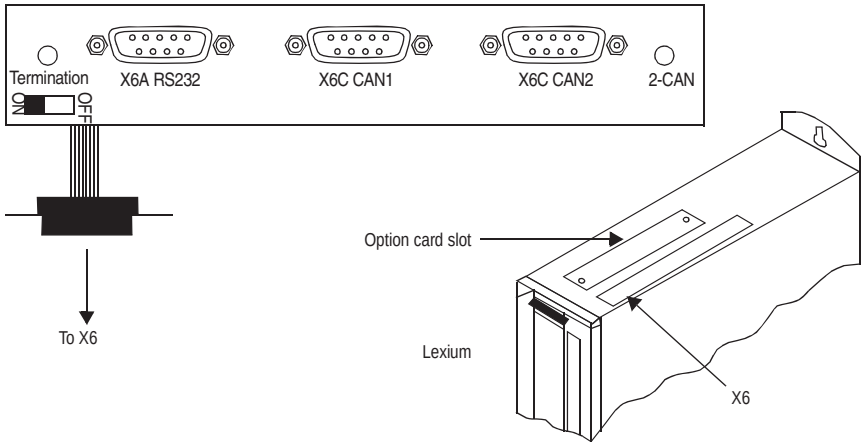
The X6 connector on Lexium 15 servo drives carries the signals for an RS232 link and for a CAN bus. This means that a special cable is needed to ensure that both the RS232 link and the CAN bus remain immediately available on the servo drive. Moreover, pin assignment on this X6 connector is not standard compliant.

## Using the AM02CAN001V000 accessory

The AM02CAN001V000 module can be used to solve the problems listed above, as it includes:

- a male SubD connector for the RS232 link,
- two SubD connectors for the CAN bus, both these connectors are cabled in parallel.

Pin assignment for the three connectors is standard compliant. The module also includes a switch to activate the end of line resistance (120 Ohm) for the CAN bus.
The figure below shows the Lexium 15 connector positions.

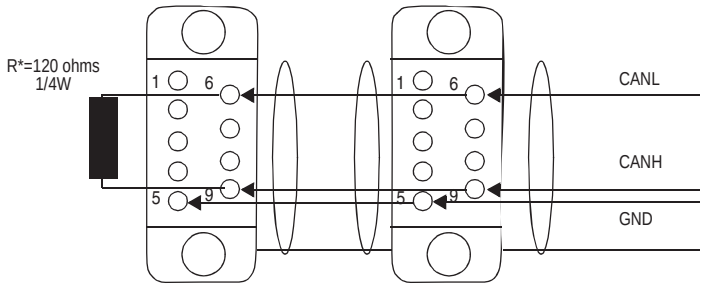



Standard cables can be used, with shielding for the RS232 and CAN interfaces.

**note:**when the servo drive is the last item on the CAN bus, the end switch must be on ON.
Otherwise, the switch must remain on OFF (delivery state).

## Without the AM02CAN001V000 accessory

If the AMO2CA001V000 extension module is not used, a SUB D 9 point female connector is required.
The figure below shows the SUB D 9 connector cabling.




Adapting the network:

- according to ISO 11898, the bus should have an impedance of 120 ohms.

The line length usable for secure communications decreases as the transfer speed increases.

# CANOpen wiring

## Bus cable specifications

In accordance with ISO 11898, you should use a bus cable with a characteristic impedance of 120 Ω . The usable cable length for reliable communication is reduced as the transmission rate is increased.

The following values that we have measured can be used as a guide. They should not, however, be interpreted as limiting values:

| Cable data | |
|---|---|
| Characteristic impedance | 100 ... 120 Ω |
| Cable capacitance | Max 60 nF/km |
| Lead resistance (loop) | 159.8 Ω/km |

The following table gives the cable length as a function of the transmission rate:

| Transmission rate: kbit/s | Lexium 15 LP - Max. cable length: m | Lexium 15 MP/HP - Max. cable length: m |
|---|---|---|
| 1000 | 20 | 20 |
| 500 | 35 | 70 |
| 250 | 50 | 115 |

Longer transmission distances may be achieved with a lower cable capacitance (max. 30 nF/km) and lower lead resistance (loop, 115 Ω/km).
(Characteristic impedance 150 ± 5 Ω -> termination resistance 150 ± 5 Ω ).

For EMC reasons, the SubD connector housing must fulfill the following requirements:

- metal or metallic housing,
- provision for connecting the cable shielding in the housing, with a large contact area (metallic housing).

The entire cabling system to be used corresponds to the CAN standard.

## Standards

Lexium 15 CANopen implementation is close to DS 301V4.02 and DSP 402 V2.0 profile. However, it is not fully compliant. If you need specific details, please contact Schneider Electric technical support.

For the manufacturer functions we will introduce the list of CANopen objects equivalent to ASCII commands. The function definitions are given in the Lexium 15 Programming Manual.

The Objects to access to ASCII commands have 4 sub indexes:

- 1: to access to the ASCII command value or execute the command.
- 2: lower limit of the value.
- 3: upper limit of the value.
- 4: default value.

# CANopen overview

## Device data structure

The device data structure is as follow:

- Each device has a set of parameters (gain, acceleration, communication parameters...).
- Each parameter is identified by an index and a sub index number and the ability to be mapped or not mapped in a PDO.

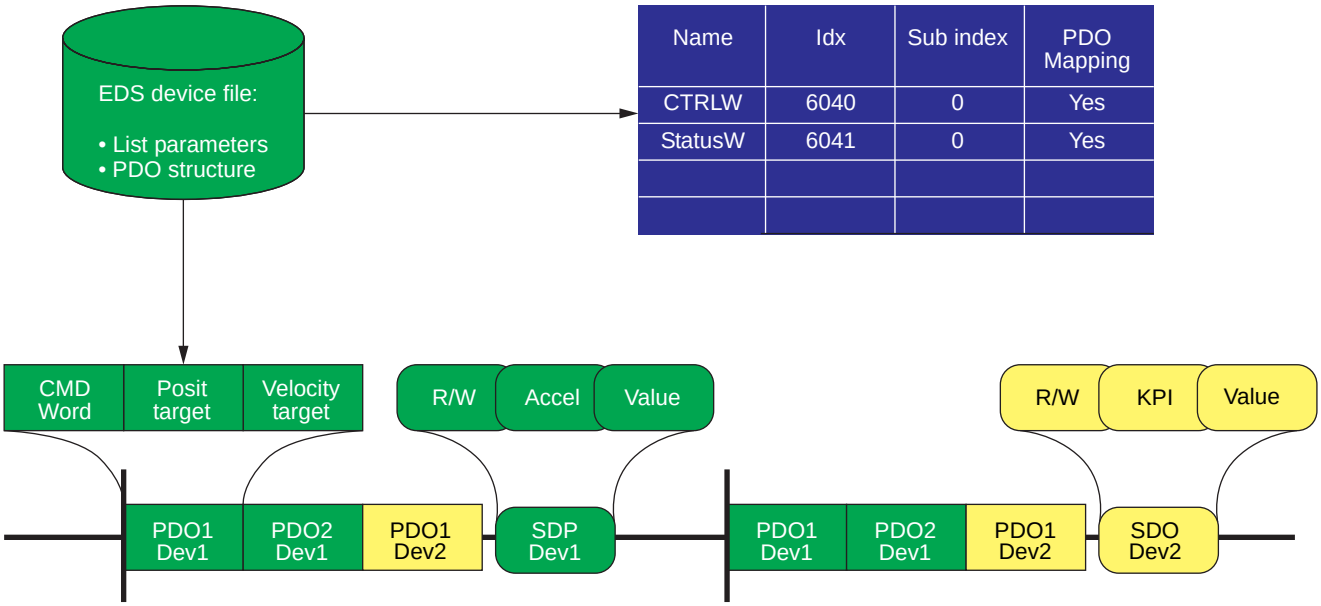
## Types of communication

There are two types of communication between devices:

- PDO Process Data Object. They are standardized communication objects dedicated for parameters cyclic exchange.
- SDO Service Data Object. They are configuration data dedicated for parameters acyclic exchange.

## Highlights of CANopen

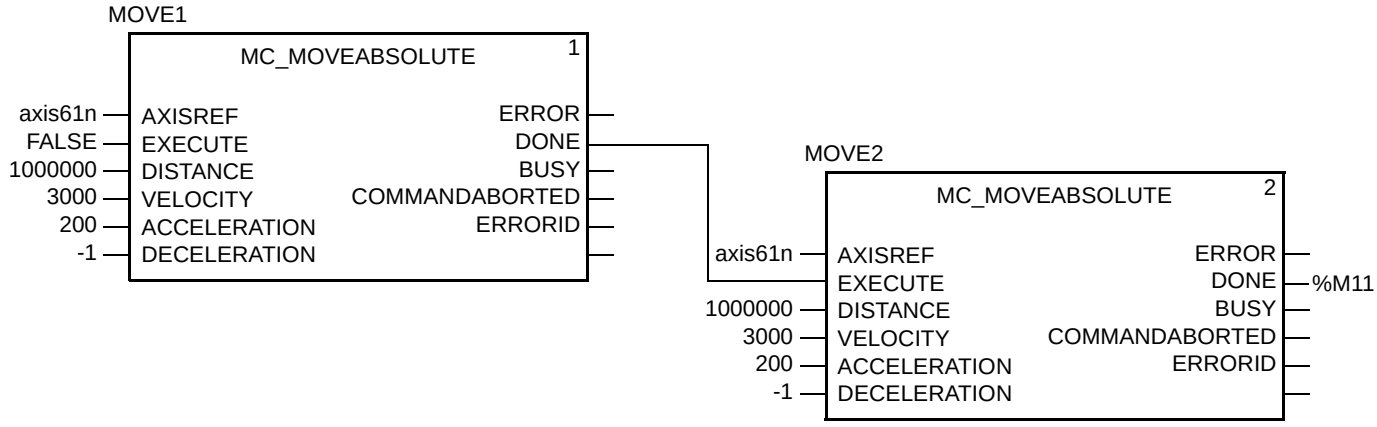
The diagram below is a brief introduction of CANopen.




# PLCopen Motion Function Blocks for Schneider Motion Controllers

You do not need to manage this notion with =S= controllers. In fact PLCopen Motion Function Bloc libraries are available for:

- Premium PLC with P-Unit programming software since Dec 2005.
- BL Motion Controller with CoDeSys programming software since March 2005.
- Mirano PLC with P-Unit programming software coming soon.




**The information below is dedicated to persons who use other =S= controllers with PL7 programming language or third party controllers with basic CANopen functions.**

## PDO and SDO notions

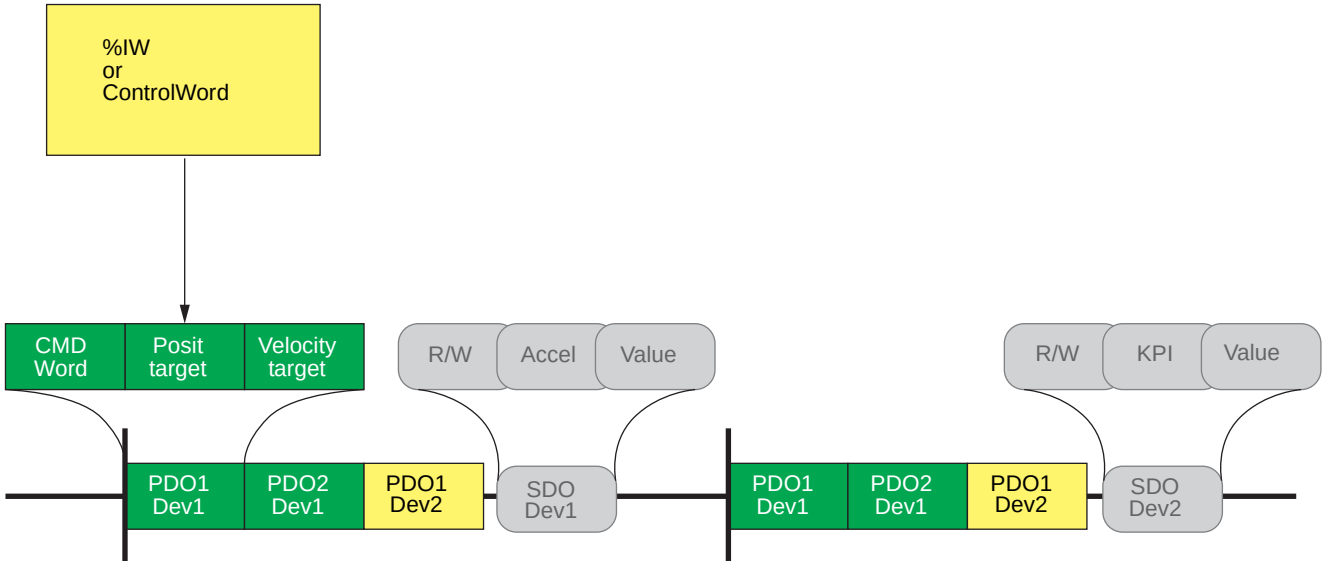
The access to objects exchanged by PDO in IEC 61131 languages may be made in different ways.

### CANopen and IEC 61131 for PDO

Depending on the implementation, the objects mapped in PDS are automaically refreshed by the system. They can be read or writen by IEC61131 objects such as:

- Generic names, like %IW or %QW. Those objects can be implemented by :
  - "Raw" objects or objects mapped in series of words. In this case, a 32 bits word will be stored in two IEC61131 words. For instance, the Target Position object which is a 32 bits piece of information, will be stored in PL7 in 2 %QW objects.
  - Type oriented mapping objects. In that case, each PDO object corresponds to an IEC61131 object. In the same example than previously, the Target position object will be stored in a %QDW object. This is the P-unit approach
- Symbolic names: Objects will be stored in IEC61131 objects with necessary size. In the previous example , Target Position will be stored in  a 32 bits IEC61131 object. This is the P-Unit or CoDeSys approach.

The following diagram is a brief introduction to SDO and PDO notions



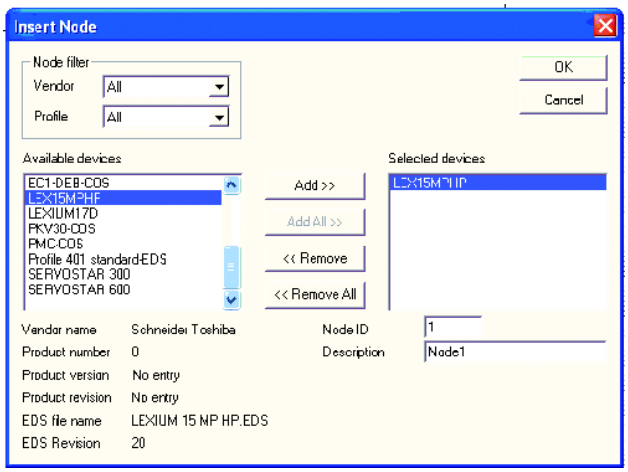
## PDO mapping configuration

The configuration of PDO is made by CANOpen communication profile configurator. Follow the next steps to achieve this PDO mapping configuration:

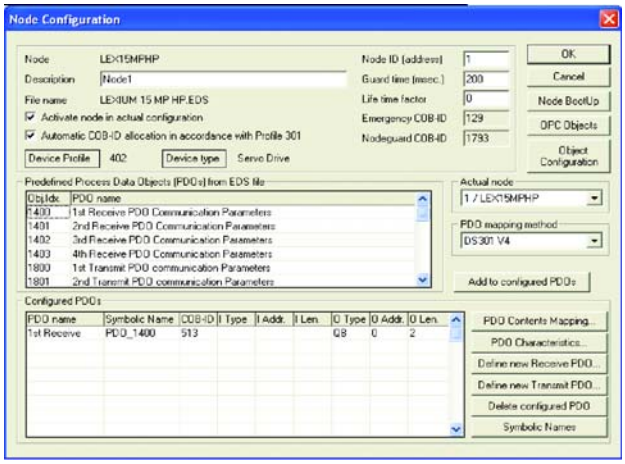
1) Copy EDS files which are on the CD ROM that comes with the product in C:\Program Files\Hilscher\SyCon\Fieldbus\CANopen\EDS folder.

2) Copy .dib files which are on the CD ROM that comes with the product in C:\Program Files\Hilscher\SyCon\Fieldbus\CANopen\BMP folder.

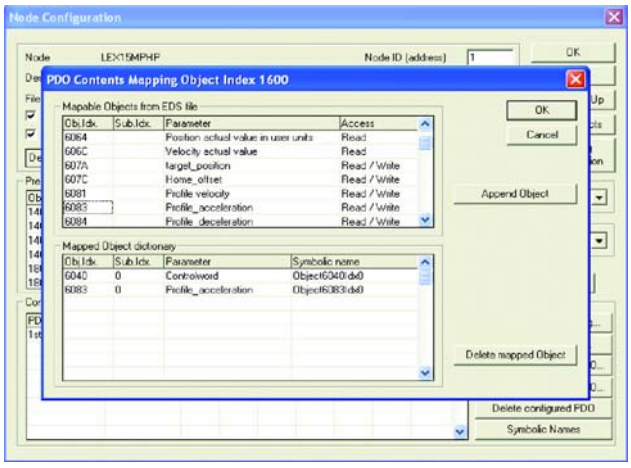
3) Select Lexium15L or Lexium15MH device for Lexium 15 LP and Lexium 15 MP/HP.




4) Select a predefined PDO. In the example below the selected PDO has Controlword and Digital Speed or current predefined:



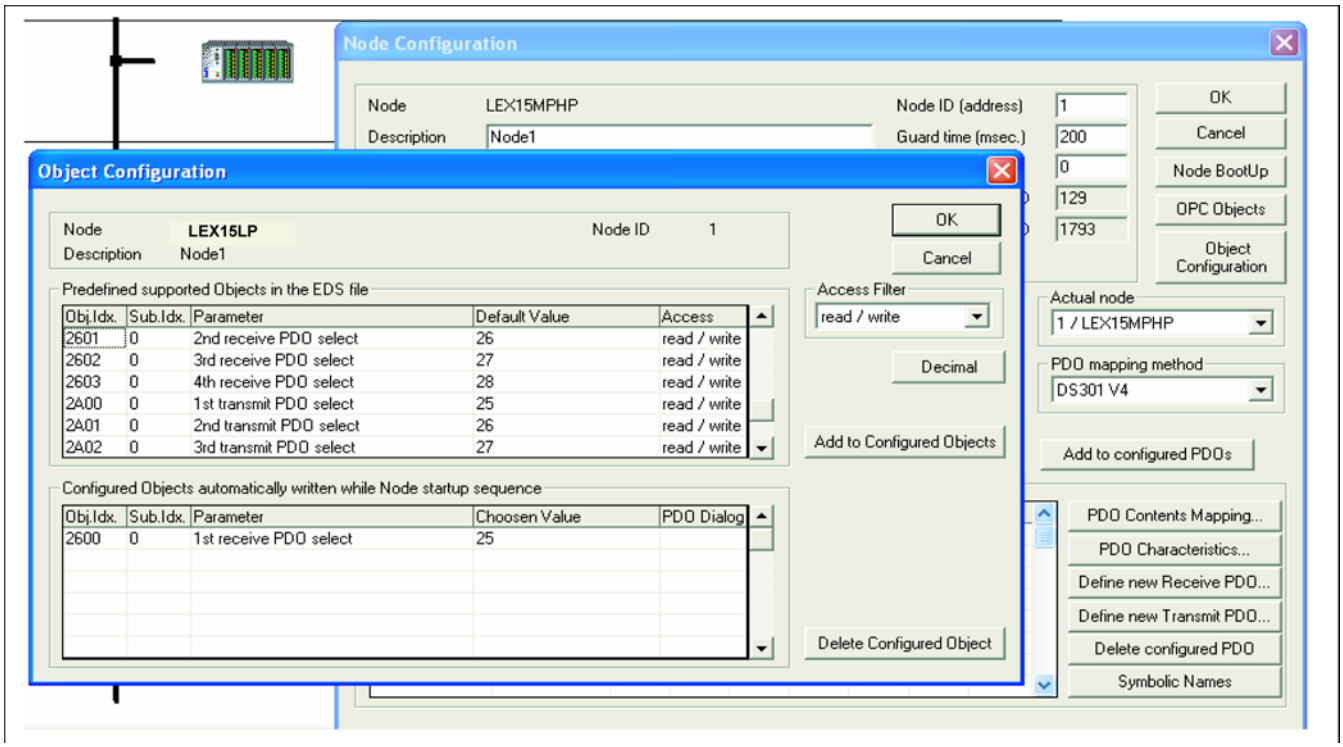

5) You can remove or add objects from the predefined PDO. In the example below the Profile_acceleration object has been added:



## Lexium 15 predefined PDO

All Lexium 15 PDOs are free mapping. Nevertheless the EDS file provides predefined PDOs recommended by DSP 402 norm. This mapping is available for Lexium 15 LP/MP/HP, however Lexium 15 MP and Lexium 15 HP need to include objects 2600h to 2603h and 2A00h to 2A03h.

The following print screen shows how to add those objects




The tables below describe the transmit predefined PDOs:

| PDO Number | Mapping Object Index | Mapping Object Name | Lex 15 LP | Lex 15 MP/HP | Lex 15 LP | Lex 15 MP/HP |
|---|---|---|---|---|---|---|
| | | | Implementation | | Free Mapping | |
| 1 | 6041h | Statusword | Yes | Yes | Yes | Yes |
| 2 | 6041h | Statusword | No | No | Yes | Yes |
| | 6061h | Modes of operation display | | | | |
| 3 | 6041h | Statusword (pp) | No | No | Yes | Yes |
| | 6064h | Position actual value | | | | |
| 4 | 6041h | Statusword (pv) | No | No | Yes | No |
| | 606Ch | Velocity actual value | | | | |

### PLEASE NOTE :

Ensure that BUSP7 ASCII command is set to 1. Another value of this ASCII will cause wrong value display in the objects 6063h and 6064h.

The table below describes the received predefined PDOs:

| PDO Number | Mapping Object Index | Mapping Object Name | Lex 15 LP | Lex 15 MP/HP | Lex 15 LP | Lex 15 MP/HP |
|---|---|---|---|---|---|---|
| | | | Implementation | | Free Mapping | |
| 1 | 6040h | Controlword | Yes | Yes | Yes | Yes |
| 2 | 6040h | Controlword | Yes | No | Yes | Yes |
| | 6060h | Modes of operation | | | | |
| 3 | 6040h | Controlword | Yes | No | Yes | Yes |
| | 607Ah | Target position | | | | |
| 4 | 6040h | Controlword | Yes | No | Yes | Yes |
| | 60FFh | Target velocity (pv) | | | | |

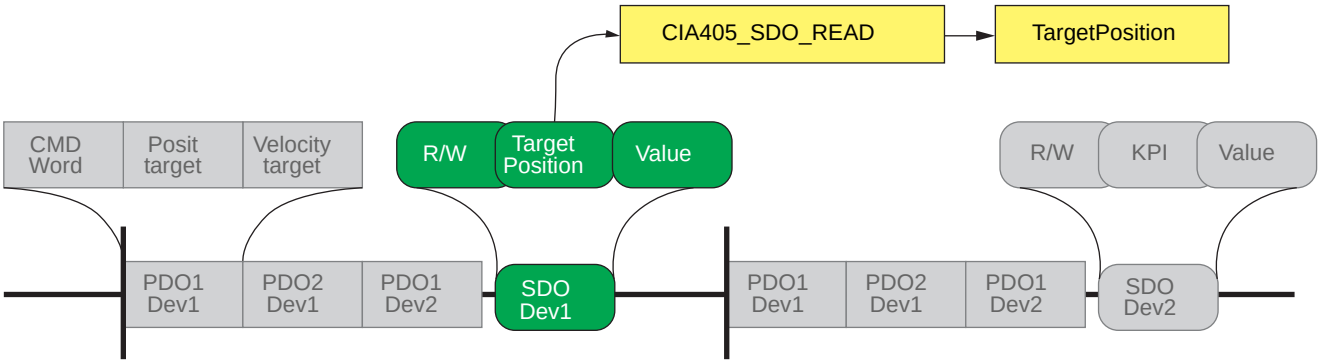
## SDO handling

The DS 405 profile gives the recommendation for SDO objects access implementation:
All CANopen objects may be accessed by SDO exchanges. The DS405 implementation recommendation offer two main functions:

- CIA405_SDO_READx
- CIA405_SDO_WRITEx

These functions enable to store information in IEC 61131 Symbolic named object. With those functions, it is possible to read/write CANopen objects into/from IEC 61131 symbolic named objects i.e read or write the Target Position.

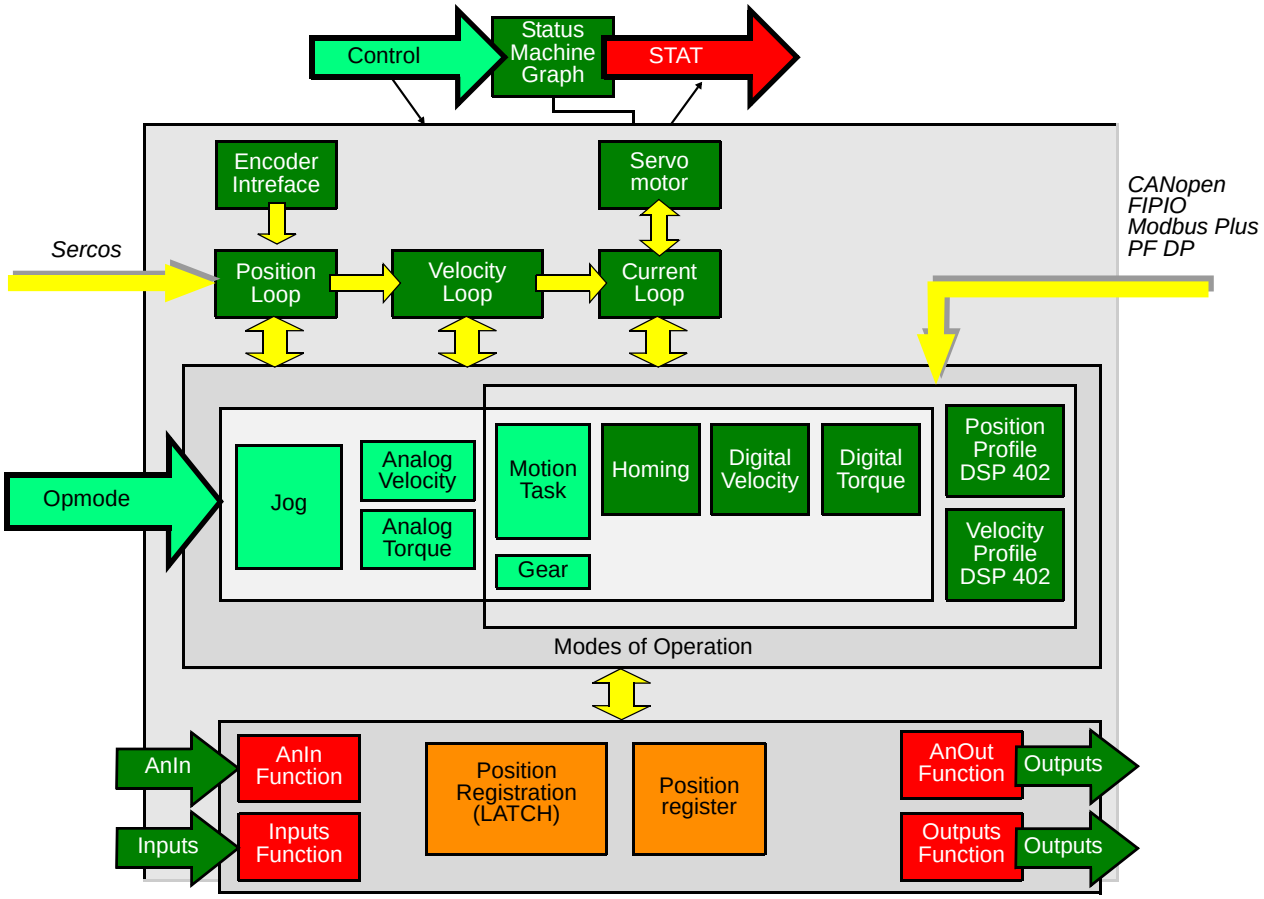
The following diagram describes the SDO handling




**note:**Other IEC 61131 implementations enable similar SDO handling, only the syntax is different

# Device overview

The diagram below describes the Lexium 15 servo drive function structure:



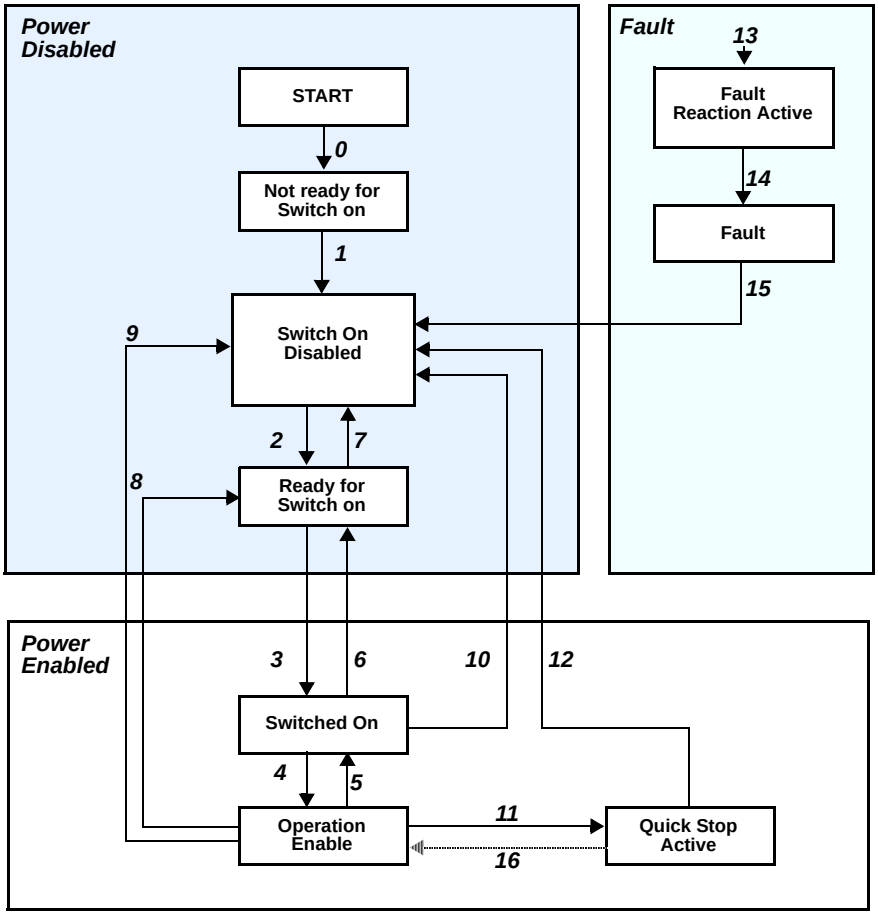

The Lexium 15 servo drive has been structured in several functions:

- DRVCOM to manage the Status Machine Graph (Power ON / Power OFF / Fault).
- Modes of Operations to execute Motion Functions.
- Smarts functions to handle Digital I/O and Analog I/O.
- Position Register.
- Position registration (Latch).

All these functions can be managed by CANopen. In the next chapters we will explain how to handle them using CANopen communication profile.

## Device Control Graph for Lexium 15 LP

The device state graph implemented by Lexium 15LP servo drives is as follow:




The following table lists the different states of the Status Graph

| State | Description |
|---|---|
| Not Ready for Switch On | Lexium 15 LP is not ready for switch on, there is no operational readiness (R1A/R1C) signaled from the controller program. |
| Switch On Disable | Lexium 15 LP is ready for switch on, parameters can be transferred, the DC-link voltage can be switched on, motion functions cannot be carried out yet. |
| Ready for Switch On | DC-link voltage may be switched on, parameters can be transferred, motion functions cannot be carried out yet. |
| Switched On | DC-link voltage must be switched on, parameters can be transferred, motion func-tions cannot be carried out yet, output stage is switched on (enabled). |
| Operation Enable | No fault present, output stage is enabled, motion functions are enabled. |
| Quick Stop Active | Servo drive has been stopped with the emergency ramp, output stage is enabled, motion functions are enabled, response depends on SDO 605Ah. |
| Fault Reaction Active | A fault has occured and the servo drive is stopped with the quickstop ramp. |
| Fault | A fault is active, the servo drive has been stopped and disable. |

## State transitions of the servo drive supervisor

The state transitions are affected by internal events (e.g. switching off the DC-link voltage) and by the flags in the Control Word (bits 0,1,2,3,7). The following table describes the State transitions.

| Transition | Event | Action |
|---|---|---|
| 0 | Reset | Initialization. |
| 1 | Initialization completed successfully.<br>Lexium 15 LP is ready to operate. | None. |
| 2 | Bit 1 *Disable Voltage* and Bit 2 *Quick Stop* are set in the Control word (*Shut-down* command). DC-link voltage may be present. | None. |
| 3 | Bit 0 is also set (*Switch On* command). | Output stage is switched on (enabled), provided that the hardware enable is present (logical AND). Servo drive has torque. |
| 4 | Bit 3 is also set (*Enable Operation* command). | Motion function is enabled, depending on the mode that is set. |
| 5 | Bit 3 is canceled (*Disable Operation* command). | Motion function is inhibited. Servo drive is stopped, using the relevant ramp (mode-dependent). The present position is maintained. |
| 6 | Bit 0 is canceled (*Shutdown* command). | Output stage is disabled. Servo drive has no torque. |
| 7 | Bits 1 and 2 are canceled (*Quick Stop* / *Disable Voltage* command). | None. |
| 8 | Bit 0 is canceled (*Shutdown* command). | Output stage is disabled. Servo drive has no torque. |
| 9 | Bit 1 is canceled (*Disable Voltage* command). | Output stage is disabled. Servo drive has no torque. |
| 10 | Bits 1 and 2 are canceled (*Quick Stop* / *Disable Voltage* command). | Motion function is enabled, depending on the mode that is set. |
| 11 | Bit 2 is canceled (*Quick Stop* command). | Servo drive is stopped with the emergency braking ramp. The output stage remains enabled. Setpoints are canceled (motion block number, digital setpoint, speed for jogging or homing). Bit 2 must be set again before any further motion tasks can be performed. |
| 12 | Bit 1 is canceled (*Disable Voltage* command). | Output stage is disabled. Servo drive has no torque. |
| 13 | Fault reaction active. | Execute appropriate fault reaction . |
| 14 | Fault reaction is completed. | Servo drive function is disabled. The power section may be switched off. |
| 15 | *Fault Reset* command received from host. | A reset of the fault condition is carried out if no fault exists currently on the servo drive. After leaving the state Fault the Bit 7 'Reset Fault' of the Controlword has to be cleared by the host. |
| 16 | Bit 2 is set. | Motion function is enabled again. |

**⚠ CAUTION**

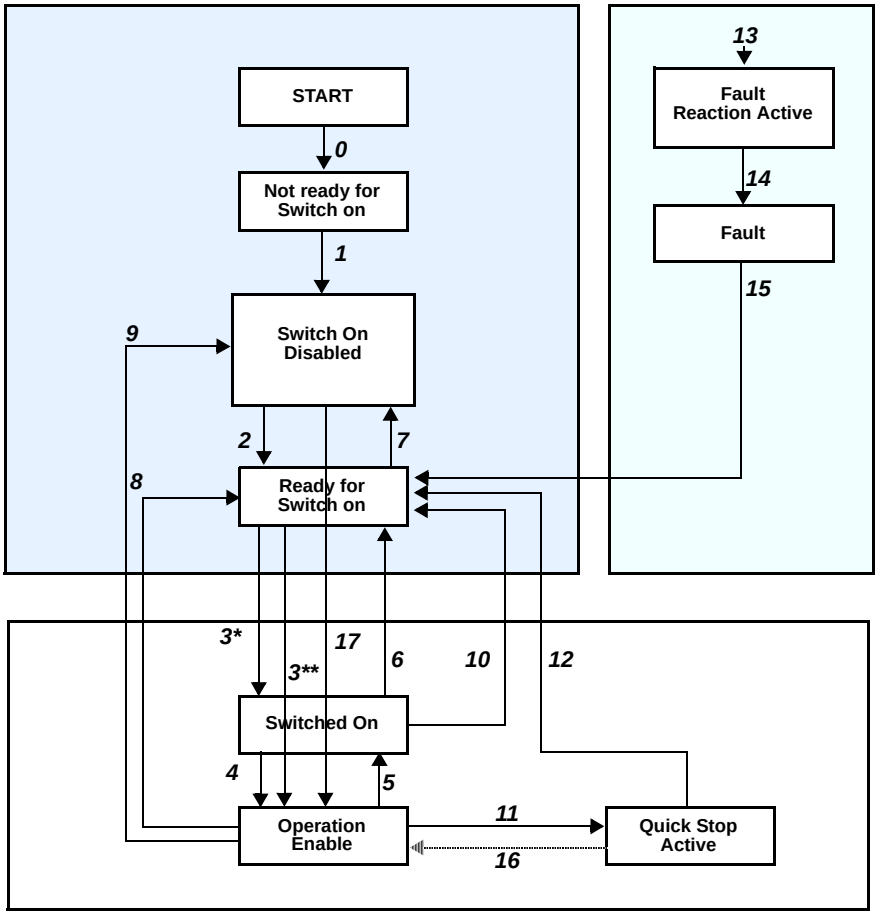
**INJURY FROM UNEXPECTED MOVEMENTS**

If the servo drive is operated through the Control word / Status word, then no control commands may be sent through another communication channel (RS232, CANopen, ASCII channel, Option board).

**Failure to follow this instruction can result in injury or equipment damage.**

## Device Control Graph for Lexium 15 MP/HP

The device state graph implemented by Lexium 15 MP/HP servo drives is as follow:




The following table lists the different states of the Status Graph

| State | Description |
|---|---|
| Not Ready for Switch On | Lexium 15 MP/HP is not ready for switch on, there is no operational readiness (R1A/R1C) signaled from the controller program. |
| Switch On Disable | Lexium 15 MP/HP is ready for switch on, parameters can be transferred, the DC-link voltage can be switched on, motion functions cannot be carried out yet. |
| Ready to Switch On | DC-link voltage may be switched on, parameters can be transferred, motion functions cannot be carried out yet. |
| Switched On | DC-link voltage must be switched on, parameters can be transferred, motion functions cannot be carried out yet, output stage is switched on (enabled). |
| Operation Enable | No fault present, output stage is enabled, motion functions are enabled. |
| Quick Stop Active | Servo drive has been stopped with the emergency ramp, output stage is enabled, motion functions are enabled, response depends on SDO 605Ah. |
| Fault Reaction Active | Not supported at present. |
| Fault | Not supported at present. |

## State transitions of the servo drive supervisor

The state transitions are affected by internal events (e.g. switching off the DC-link voltage) and by the flags in the Control Word (bits 0,1,2,3,7). The following table describes the State transitions.

| Transition | Event | Action |
|---|---|---|
| 0 | Reset | Initialization. |
| 1 | Initialization completed successfully. Lexium 15 MP/ HP is ready to operate. | None. |
| 2 | Bit 1 *Disable Voltage* and Bit 2 *Quick Stop* are set in the Control word (*Shut-down* command). DC-link voltage may be present. | None. |
| 3* | Bit 0 is also set (*Switch On* command). | Output stage is switched on (enabled), provided that the hardware enable is present (logical AND). Servo drive has torque. |
| 3** | Bit 3 is also set (*Enable Operation* command). | Motion function is enabled, depending on the mode that is set (Manufacturer Modes - motion function is enabled). |
| 4 | Bit 3 is also set (*Enable Operation* command). | Motion function is enabled, depending on the mode that is set. |
| 5 | Bit 3 is canceled (*Disable Operation* command). | Motion function is inhibited. Servo drive is stopped, using the relevant ramp (mode-dependent). The present position is maintained. |
| 6 | Bit 0 is canceled (*Shutdown* command). (If bit 1 in DRVCNFG is set and if HW-Enable is taken away an Emergenc y Message will result) | Output stage is disabled. Servo drive has no torque. |
| 7 | Bits 1/2 are canceled (*Quick Stop/Disable Voltage* command). | None. |
| 8 | Bit 0 is canceled (*Shutdown* command). | Output stage is disabled. Servo drive has no torque. |
| 9 | Bit 1 is canceled (*Disable Voltage* command). | Output stage is disabled. Servo drive has no torque. |
| 10 | Bits 1 and 2 are canceled (*Disable Voltage* command). (If bit 1 in DRVCNFG is set and if HW-Enable is taken away an Emergency Message will result) | Motion function is enabled, depending on the mode that is set. |
| 11 | Bit 2 is canceled (*Quick Stop* command). | Servo drive is stopped with the emergency braking ramp. The output stage remains enabled. Setpoints are canceled (motion block number, digital setpoint, speed for jogging or homing). Bit 2 must be set again before any further motion tasks can be performed. |
| 12 | Bit 1 is canceled (*Disable Voltage* command). | Output stage is disabled. Servo drive has no torque. |
| 13 | A fault has occurred in the servo drive. | Execute appropriate fault reaction, i.e. the servo drives ramps down with DECSTOP - ramp and is then disabled. |
| 14 | Fault reaction is completed. | Servo drive function is disabled. The power section may be switched off. |
| 15 | *Fault Reset* command received from host. | A reset of the fault condition is carried out if no fault exists currently on the servo drive. After leaving the state Fault the Bit 'Reset Fault' of the controlword has to be cleared by the host. |
| 16 | Bit 2 is set. | Motion function is enabled again. |
| 17 | Bit 1 *Disable Voltage* and Bit 2 *Quick Stop* and Bit 3 *Operation Enable* are set in the Control Word. | Output stage is switched on (enabled), provided that the hardware enable is present (logical AND). Motion function is enabled, depending on the mode that is set. |

**⚠ CAUTION**

**INJURY FROM UNEXPECTED MOVEMENTS**

If the servo drive is operated through the Control word / Status word, then no control commands may be sent through another communication channel (RS232, CANopen, ASCII channel, Option board).

**Failure to follow this instruction can result in injury or equipment damage.**

# Device control

## Control Word

The Control Word (Index 6040h) device controls the Status machine graph as well as the Mode of operation:

| | Bit | Name | DSP402 M/O | Profile Position Mode | Profile Velocity Mode | Profile Torque Mode | Homing Mode | Interpo -lation mode | Lex 15 LP | Lex 15 MP/HP |
|---|---|---|---|---|---|---|---|---|---|---|
| DRVCOM management | 0 | Switch on | M | | | | | | ok | ok |
| | 1 | Disable Voltage | M | | | | | | ok | ok |
| | 2 | Quick Stop | M | | | | | | ok | ok |
| | 3 | Enable Operation | M | | | | | | ok | ok |
| Modes of operation management | 4 | Operation mode specific | | New_set_point | Reserved | Reserved | Homing Operation Start | Enable_ip _mode | ok | ok |
| | 5 | Operation mode specific | | Change_set _immediately | Reserved | Reserved | Reserved | Reserved | ok | ok |
| | 6 | Operation mode specific | | 0: absolute 1: relative | Reserved | Reserved | Reserved | Reserved | ok | ok |
| | 7 | Reset Fault | M | | | | | | ok | ok |
| | 8 | Pause/halt | | | | | | | ok | ok |
| | 9 | Reserved | | | | | | | ok | ok |
| | 10 | Reserved | | | | | | | ok | ok |
| | 11 | Reserved | | | | | | | ok | Reset following error |
| | 12 | Reset position | | | | | | | ok | ok |
| | 13 | Manufacturer-specific | | | | | | | ok | ok |
| | 14 | Manufacturer-specific | | | | | | | ok | ok |
| | 15 | Manufacturer-specific | | | | | | | ok | ok |

| Command | Bit 7 Fault Reset | Bit 3 Enable Operation | Bit 2 Quick Stop | Bit 1 Disable Voltage | Bit 0 Switch on | Transitions |
|---|---|---|---|---|---|---|
| Shutdown | X | X | 1 | 1 | 0 | 2. 6. 8 |
| Switch on | X | X | 1 | 1 | 1 | 3 |
| Disable Voltage | X | X | X | 0 | X | 7. 9. 10. 12 |
| Quick Stop | X | X | 0 | 1 | X | 7. 10. 11 |
| Disable Operation | X | 0 | 1 | 1 | 1 | 5 |
| Enable Operation | X | 1 | 1 | 1 | 1 | 4. 16 |
| Fault Reset | 1 | X | X | X | X | 15 |

## Modes of operation

Lexium 15 servo drives support the following modes of operation

| Modes Of Operation | DSP 402 | Lexium 15 LP Available | Lexium 15 MP/HP Available | Unilink Number |
|---|---|---|---|---|
| **CANopen DS 402 Modes of operation** | | | | |
| No mode change/no mode assigned | 0 | yes | yes | N/A |
| Profile position mode (pp) | 1 | yes | yes | N/A |
| Velocity mode (vl) | 2 | | | |
| Profile velocity mode (pv) | 3 | yes | yes | N/A |
| Torque profile mode (tq) | 4 | yes | | |
| Reserved | 5 | | | |
| Homing mode (hm) | 6 | yes | yes | N/A |
| Interpolated position mode (ip) | 7 | yes | | |
| Cyclic sync position mode | 8 | | | |
| Cyclic sync velocity mode | 9 | | | |
| Cyclic sync torque mode | 10 | | | |
| Reserved | +11 to +127 | | | |
| **Manufacturer** | | **DSP 402 nb when available** | | |
| Electronical gearing | | F7h | F7h | 4 |
| Jogging | | F8h | F8h | |
| Homing proprietary | | | F9h | |
| Trajectory (Positioning specific mode) | | | FAh | |
| Analog Torque | | 83h | FBh | 3 |
| Analog speed | | 81h | FCh | 1 |
| Digital torque | | 82h | FDh | 2 |
| Digital speed | | 80h | FEh | 0 |
| Position (MTASK) | | FFh | FFh | 8 |
| Sercos | | | | 6 |

The object Mode Of Operation (Index: 6060h) is used to select the mode of operation of the servo drive.

The Mode Of Operation selected is displayed by the Mode Of Operation display object (Index 6061h).

## Status Word

The Status Word (Object Index 6041h) provides the information of Device Control graph status and Mode of operation status:

| | Bit | Name | Profile Position Mode | Profile Velocity Mode | Profile Torque Mode | Homing Mode | Interpo-lation Mode | Lex 15 LP | Lex 15 MP/HP |
|---|---|---|---|---|---|---|---|---|---|
| DRVCOM management | 0 | Ready to switch on | | | | | | ok | ok |
| | 1 | Switched on | | | | | | ok | ok |
| | 2 | Operation enable | | | | | | ok | ok |
| | 3 | Fault | | | | | | ok | ok |
| | 4 | Disable voltage | | | | | | ok | ok |
| | 5 | Quick stop | | | | | | ok | ok |
| | 6 | Switch on disabled | | | | | | ok | ok |
| | 7 | Warning | | | | | | ok | ok |
| | 8 | Manufacturer-specific (reserved) | | | | | | ok | ok |
| | 9 | Remote | | | | | | ok | ok |
| Modes of operation management | 10 | Target reached | Yes | Yes | | | | ok | ok |
| | 11 | Internal limit active | Yes | Yes | Yes | Yes | Yes | ok | ok |
| | 12 | Operation mode specific (reserved) | Setpoint acknowledge | Velocity zero attained | Homing attained | Reserved | Ip mode active | ok | ok |
| | 13 | Operation mode specific (reserved) | Following error | Max slippage error | Homing error | Reserved | Reserved | ok | ok |
| | 14 | Manufacturer-specific | | | | | | Gearing done | Gearing done |
| | 15 | Manufacturer-specific | | | | | | Halt req | Halt req |

## Digital inputs operations

To manage the Digital Inputs access through CANopen, you may use the following objects:

| Lexium 15 MP/HP | | | Lexium 15 LP | | | | |
|---|---|---|---|---|---|---|---|
| Status | Format | PDO mapping | Format | PDO mapping | Index | Sub | Name |
| | | | | | | | |
| - | | | UNSIGNED32 | Yes | 60FDh | 0 | Digital inputs |

### Lexium 15 LP

| Index | Sub Idx | ASCII | Access | Format | Units | Limits | Short description |
|---|---|---|---|---|---|---|---|
| 3561h | 1 | IN1 | ro | Integer8 | - | - | Status of digital input 1 |
| 3561h | 1 | IN5_20 | ro | Integer32 | - | long int | Status of digital inputs 5 ... 20 |
| 3562h | 1 | IN1MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Input 1 |
| 3562h | 1 | IN5_20MODE | rw | Integer32 | - | long int | Function of digital inputs 5 ... 20 |
| 3563h | 1 | IN1TRIG | rw | Integer32 | - | long int | Variable for IN1MODE |
| 3563h | 1 | IN5_20TRIG | rw | Integer32 | - | long int | Variable for digital inputs 5 ... 20 |
| 3564h | 1 | IN2 | ro | Integer8 | - | - | Status of digital input 2 |
| 3565h | 1 | IN2MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Input 2 |
| 3566h | 1 | IN2TRIG | rw | Integer32 | - | long int | Variable for IN2MODE |
| 3567h | 1 | IN3 | ro | Integer8 | - | - | Status of Digital Input 3 |
| 3568h | 1 | IN3MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Input 3 |
| 3569h | 1 | IN3TRIG | rw | Integer32 | - | long int | Variable for IN3MODE |
| 356Ah | 1 | IN4 | ro | Integer8 | - | - | Status of Digital Input 4 |
| 356Bh | 1 | IN4MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Input 4 |
| 356Ch | 1 | IN4TRIG | rw | Integer32 | - | long int | For IN4MODE |

### Lexium 15 MP/HP

| Index | Sub Idx | ASCII | Access | Format | Units | Limits | Short description |
|---|---|---|---|---|---|---|---|
| 3561h | 1 | IN1 | ro | Integer8 | - | - | Status of digital input 1 |
| 3562h | 1 | IN1MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Input 1 |
| 3563h | 1 | IN1TRIG | rw | Integer32 | - | long int | Variable for IN1MODE |
| 3564h | 1 | IN2 | ro | Integer8 | - | - | Status of digital input 2 |
| 3565h | 1 | IN2MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Input 2 |
| 3566h | 1 | IN2TRIG | rw | Integer32 | - | long int | Variable for IN2MODE |
| 3567h | 1 | IN3 | ro | Integer8 | - | - | Status of Digital Input 3 |
| 3568h | 1 | IN3MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Input 3 |
| 3569h | 1 | IN3TRIG | rw | Integer32 | - | long int | variable for IN3MODE |
| 356Ah | 1 | IN4 | ro | Integer8 | - | - | Status of Digital Input 4 |
| 356Bh | 1 | IN4MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Input 4 |
| 356Ch | 1 | IN4TRIG | rw | Integer32 | - | long int | Variable for IN4MODE |

## Digital Outputs operations

To manage the Digital Outputs access through CANopen, you may use the following objects:

### Lexium 15 LP

| Index | Sub Index | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 3534h | 1 | ENCMODE | rw | Integer8 | - | 0, 1, 2, 3 | Selection of Encoder Emulation |
| 3535h | 1 | ENCOUT | rw | Integer32 | CPR | see ASCII command description | Resolution Encoder Emulation EEO (ROD) |
| 3537h | 1 | ENCZERO | rw | Integer16 | - | 0 ... ENCOUT-1 | Zero Pulse Offset EEO (ROD) |
| 35AEh | 1 | O1 | rw | Integer8 | - | - | State of Digital Output 1 |
| 35AEh | 1 | O3_18 | rw | Integer32 | - | long int | State of Digital Output 1 |
| 35AFh | 1 | O1MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Output 1 |
| 35AFh | 1 | O3_18MODE | rw | Integer32 | - | long int | Function of Digital Output 1 |
| 35B0h | 1 | O1TRIG | rw | Integer32 | - | long int | Auxiliary variable for O1MODE |
| 35B0h | 1 | O3_18TRIG | rw | Integer32 | - | long int | Auxiliary variable for O1MODE |
| 35B1h | 1 | O2 | rw | Integer8 | - | - | State of Digital Output 2 |
| 35B2h | 1 | O2MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Output 2 |
| 35B3h | 1 | O2TRIG | rw | Integer32 | - | long int | Auxiliary variable for O2MODE |

### Lexium 15 MP/HP

| Index | Sub Index | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 3530h | 1 | EN | Cmd | - | - | - | Software-Enable |
| 3534h | 1 | ENCMODE | rw | Integer8 | - | 0, 1, 2, 3 | Selection of Encoder Emulation |
| 3535h | 1 | ENCOUT | rw | Integer32 | CPR | see ASCII command description | Resolution Encoder Emulation EEO (ROD) |
| 3537h | 1 | ENCZERO | rw | Integer16 | - | 0 ... ENCOUT-1 | Zero Pulse Offset EEO (ROD) |
| 35AEh | 1 | O1 | rw | Integer8 | - | - | State of Digital Output 1 |
| 35AFh | 1 | O1MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Output 1 |
| 35B0h | 1 | O1TRIG | rw | Integer32 | - | long int | Auxiliary variable for O1MODE |
| 35B1h | 1 | O2 | rw | Integer8 | - | - | State of Digital Output 2 |
| 35B2h | 1 | O2MODE | rw | Integer8 | - | 0 ... 50 | Function of Digital Output 2 |
| 35B3h | 1 | O2TRIG | rw | Integer32 | - | long int | Auxiliary variable for O2MODE |
| 362Eh | 1 | IN2PM | rw | Integer8 | - | 0, 1, 2 | In-Position 2 Mode |

## Analog input operations

To manage the Analog Inputs access through CANopen, you may use the following objects:

### Lexium 15 LP

| Index | Sub Idx | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 3507h | 1 | ANCNFG | rw | Integer8 | - | 0 ... 14 | Configuration of Analog Input |
| 3508h | 1 | ANDB | rw | Float | Millivolts | 0.0 ... 10000.0 | Dead Band of the Analog Velocity Input Signal |
| 3509h | 1 | ANIN1 | ro | Integer32 | Millivolts | - 10000 ... 10000 | Voltage at Analog Input 1 |
| 350Ah | 1 | ANIN2 | ro | Integer32 | Millivolts | - 10000 ... 10000 | Voltage at Analog Input 2 |
| 350Bh | 1 | ANOFF1 | rw | Integer16 | Millivolts | - 10000 ... 10000 | Analog Offset for analog input 1 |
| 350Ch | 1 | ANOFF2 | rw | Integer16 | Millivolts | - 10000 ... 10000 | Analog Offset for analog input 2 |
| 350Fh | 1 | ANZERO1 | Cmd | - | - | - | Zero Analog Input 1 |
| 3510h | 1 | ANZERO2 | Cmd | - | - | - | Zero Analog Input SW2 |
| 3511h | 1 | AVZ1 | rw | Float | Milliseconds | 0.2 ... 100.0 | Filter Time Constant for analog input 1 |
| 3571h | 1 | ISCALE1 | rw | Float | A/10 Volts | 0.0 ... 100.0 | Scaling of Analog Current Setpoint 1 |
| 3572h | 1 | ISCALE2 | rw | Float | A/10 Volts | 0.0 ... 100.0 | Scaling of Analog Current Setpoint 2 |
| 3629h | 1 | VSCALE1 | rw | Integer16 | ( >VUNIT ) | - 15000 ... 15000 | SW1 Velocity Scaling Factor |
| 362Ah | 1 | VSCALE2 | rw | Integer16 | ( >VUNIT ) | - 15000 ... 15000 | SW2 Velocity Scaling Factor |
| 3699h | 1 | AN11NR | rw | Integer8 | - | 0, 1, 2, 3, 4 | N°. Of INxTRIG variable, that is changed analog |
| 369Ah | 1 | AN11RANGE | rw | Integer32 | - | - 262144 ... 262143 | Range of the analog change of INxTRIG |

### Lexium 15 MP/HP

| Index | Sub Idx | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 3507h | 1 | ANCNFG | rw | Integer8 | - | 0 ... 14 | Configuration of Analog Input |
| 3508h | 1 | ANDB | rw | Float | Millivolts | 0.0 ... 10000.0 | Dead Band of the Analog Velocity Input Signal |
| 3509h | 1 | ANIN1 | ro | Integer32 | Millivolts | - 10000 ... 10000 | Voltage at Analog Input 1 |
| 350Ah | 1 | ANIN2 | ro | Integer32 | Millivolts | - 10000 ... 10000 | Voltage at Analog Input SW2 |
| 350Bh | 1 | ANOFF1 | rw | Integer16 | Millivolts | - 10000 ... 10000 | Analog Offset for analog input 1 |
| 350Ch | 1 | ANOFF2 | rw | Integer16 | Millivolts | - 10000 ... 10000 | Analog Offset for input SW2 |
| 350Fh | 1 | ANZERO1 | Cmd | - | - | - | Zero Analog Input 1 |
| 3510h | 1 | ANZERO2 | Cmd | - | - | - | Zero Analog Input SW2 |
| 3511h | 1 | AVZ1 | rw | Float | Milliseconds | 0.2 ... 100.0 | Filter Time Constant input 1 |
| 3571h | 1 | ISCALE1 | rw | Float | A/10 Volts | 0.0 ... 100.0 | Scaling of Analog Current command 1 |
| 3572h | 1 | ISCALE2 | rw | Float | A/10 Volts | 0.0 ... 100.0 | Scaling of Analog Current command 2 |
| 3629h | 1 | VSCALE1 | rw | Integer16 | ( >VUNIT ) | - 15000 ... 15000 | Analog input 1 Velocity Scaling Factor |
| 362Ah | 1 | VSCALE2 | rw | Integer16 | ( >VUNIT ) | - 15000 ... 15000 | Analog input 2 Velocity Scaling Factor |

## Analog output operations

To manage the Analog Outputs access through CANopen, you may use the following objects:

### Lexium 15 MP/HP

| Index | Sub Idx | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 350Dh | 1 | ANOUT1 | rw | Integer8 | - | 0 ... 8 | Configuration of the Analog Output 1 |
| 350Eh | 1 | ANOUT2 | rw | Integer8 | - | 0 ... 8 | Source of the Analog Output 2 |
| 3520h | 1 | DAOFFSET1 | rw | Integer16 | Counts | 0 ... 2500 | Analog Offset Output 1 |
| 3521h | 1 | DAOFFSET2 | rw | Integer16 | Counts | 0 ... 2500 | Analog Offset Output 2 |
| 359Ah | 1 | MONITOR1 | ro | Integer16 | mV | - 10000 ... 10000 | Monitor 1 Output voltage |
| 359Bh | 1 | MONITOR2 | ro | Integer16 | mV | - 10000 ... 10000 | Monitor 2 Output Voltage |

## Registration (LATCH)

An internal function called LATCH is available to register a position; the value is stored depending on the configuration of the Digital Input

### Lexium 15 LP

| Index | Sub Idx | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 3578h | 1 | LATCH2P16 | rw | Integer16 | - | - | Latched 16-bit Position (positive edge) |
| 3579h | 1 | LATCH2N16 | rw | Integer16 | - | - | Latched 16-bit Position (negative edge) |
| 357Ah | 1 | LATCH2P32 | rw | Integer16 | - | - | Latched 32-bit Position (positive edge) |
| 357Bh | 1 | LATCH2N32 | rw | Integer16 | - | - | Latched 32-bit Position (negative edge) |
| 357Ch | 1 | LATCH1P32 | rw | Integer16 | - | - | Latched 32-bit Position (positive edge) |
| 357Dh | 1 | LATCH1N32 | rw | Integer16 | - | - | Latched 32-bit Position (negative edge) |
| 367Fh | 1 | LATCH1P16 | rw | Integer16 | - | - | Latched 16-bit Position (positive edge) |
| 3680h | 1 | LATCH1N16 | rw | Integer16 | - | - | Latched 16-bit Position (negative edge) |
| 3681h | 1 | EXTLATCH | rw | Integer8 | - | 0 ... 2 | Selection of the Source of the Latch Inputs |

### Lexium 15 MP/HP

The function must be enabled using the object 2026h sub index 1 set to 1

| Index | Sub Idx | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 3578h | 1 | LATCH16 | rw | Integer16 | - | - | Latched 16-bit Position (positive edge) |
| 3579h | 1 | LATCH16N | rw | Integer16 | - | - | Latched 16-bit Position (negative edge) |
| 357Ah | 1 | LATCH32 | rw | Integer32 | - | - | Latched 32-bit Position (positive edge) |
| 357Bh | 1 | LATCH32N | rw | Integer32 | - | - | Latched 32-bit Position (negative edge) |
| 357Ch | 1 | LATCHX32 | rw | Integer32 | - | - | Latched External 32-bit Position (positive edge) |
| 357Dh | 1 | LATCHX32N | rw | Integer32 | - | - | Latched External 32-bit Position (negative edge) |

## Position register

To access the Position Register through CANopen, you may use the following objects.
Remember that this internal register manages the axis software limits and the use of Digital Output as a switch cam.

### Lexium 15 LP

| Index | Sub Idx | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 3600h | 1 | SWCNFG | rw | Unsigned16 | - | 0 ... 65536 | Configuration of software limit switches |
| 3604h | 1 | SWE1 | rw | Integer32 | - | long int | SW limit switch (smalest position) |
| 3606h | 1 | SWE2 | rw | Integer32 | - | long int | SW limit switch (biggest position) |
| 3636h | 1 | WPOS | ro | Integer8 | - | 0, 1, 2 | Enable Position Registers |
| 363Fh | 1 | WPOSE | ro | Integer32 | - | 0 ... 65535 | Enable Fast Position Registers 1 ... 16 |
| 3640h | 1 | WPOSP | rw | Integer32 | - | 0 ... 65535 | Polarity of Fast Position Registers 1 ... 16 |
| 3641h | 1 | WPOSX | rw | Integer32 | - | 0 ... 65535 | Mode of Fast Position Registers 1 ... 16 |
| 3643h | 1 | POSRSTAT | rw | Integer32 | - | - | Status of Fast Position Registers 1 ... 16 |
| 3644h | 1 | P1 | rw | Integer8 | - | 0 ... 3 | P1 Fast Position Register |
| 3645h | 1 | P2 | rw | Integer32 | - | long int | P2 Fast Position Register |
| 3646h | 1 | P3 | rw | Integer32 | - | long int | P3 Fast Position Register |
| 3647h | 1 | P4 | rw | Integer32 | - | long int | P4 Fast Position Register |
| 3648h | 1 | P5 | rw | Integer32 | - | long int | P5 Fast Position Register |
| 3649h | 1 | P6 | rw | Integer32 | - | long int | P6 Fast Position Register |
| 364Ah | 1 | P7 | rw | Integer32 | - | long int | P7 Fast Position Register |
| 364Bh | 1 | P8 | rw | Integer32 | - | long int | P8 Fast Position Register |
| 364Ch | 1 | P9 | rw | Integer32 | - | long int | P9 Fast Position Register |
| 364Dh | 1 | P10 | rw | Integer32 | - | long int | P10 Fast Position Register |
| 364Eh | 1 | P11 | rw | Integer32 | - | long int | P11 Fast Position Register |
| 364Fh | 1 | P12 | rw | Integer32 | - | long int | P12 Fast Position Register |
| 3650h | 1 | P13 | rw | Integer32 | - | long int | P13 Fast Position Register |
| 3651h | 1 | P14 | rw | Integer32 | - | long int | P14 Fast Position Register |
| 3652h | 1 | P15 | rw | Integer32 | - | long int | P15 Fast Position Register |
| 3653h | 1 | P16 | rw | Integer32 | - | long int | P16 Fast Position Register |
| 36C7h | 1 | OS1 | rw | Integer8 | - | 0,1 | Set/Reset of “Posreg1" of the I/O Option Card |
| 36C8h | 1 | OS2 | rw | Integer8 | - | 0,1 | Set/Reset of ”Posreg2" of the I/O Option Card |
| 36C9h | 1 | OS3 | rw | Integer8 | - | 0,1 | Set/Reset of “Posreg3" of the I/O Option Card |
| 36CAh | 1 | OS4 | rw | Integer8 | - | 0,1 | Set/Reset of ”Posreg4" of the I/O Option Card |
| 36CBh | 1 | OS5 | rw | Integer8 | - | 0,1 | Set/Reset of “Posreg5" of the I/O Option Card |

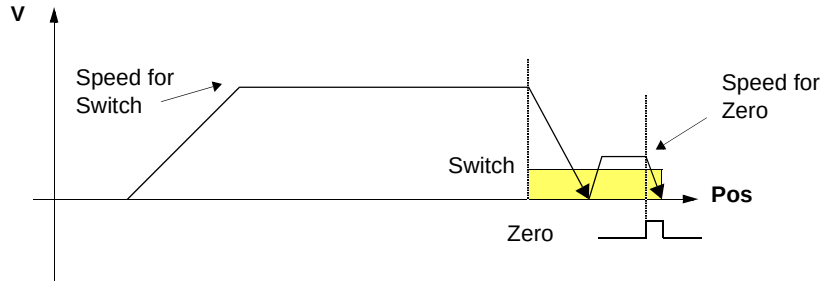
Lexium 15 MP/HP

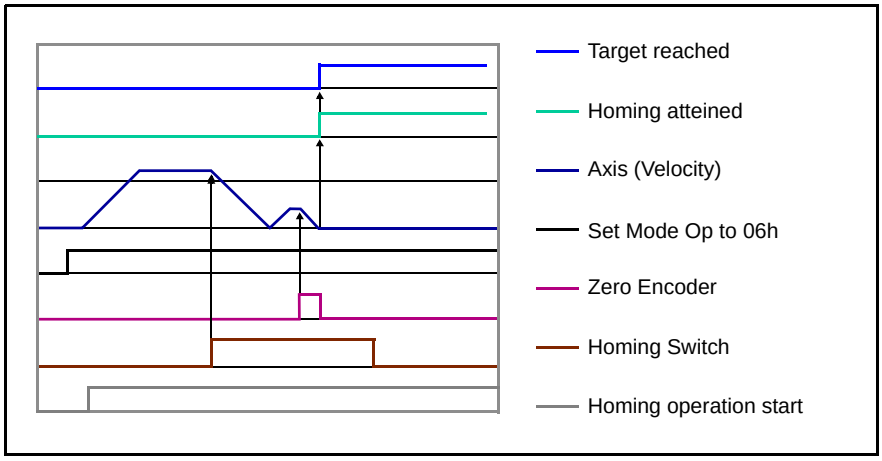
| Index | Sub Index | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 3600h | 1 | SWCNFG | rw | Unsigned16 | - | 0 ... 65536 | Configuration of Position Registers 1 ... 4 |
| 3601h | 1 | SWCNFG2 | rw | Unsigned16 | - | 0 ... 65535 | Configuration of Position Registers 0 and 5 |
| 3602h | 1 | SWE0 | rw | Integer32 | - | long int | Position register 0 |
| 3603h | 1 | SWE0N | rw | Integer32 | - | long int | Position register 0 (Cam) |
| 3604h | 1 | SWE1 | rw | Integer32 | - | long int | Position register 1 |
| 3605h | 1 | SWE1N | rw | Integer32 | - | long int | Position register 1 (Cam) |
| 3606h | 1 | SWE2 | rw | Integer32 | - | long int | Position register 2 |
| 3607h | 1 | SWE2N | rw | Integer32 | - | long int | Position register 2 (Cam) |
| 3608h | 1 | SWE3 | rw | Integer32 | - | long int | Position register 3 |
| 3609h | 1 | SWE3N | rw | Integer32 | - | long int | Position register 3 (Cam) |
| 360Ah | 1 | SWE4 | rw | Integer32 | - | long int | Position register 4 |
| 360Bh | 1 | SWE4N | rw | Integer32 | - | long int | Position register 4 (Cam) |
| 360Ch | 1 | SWE5 | rw | Integer32 | - | long int | Position register 5 |
| 360Dh | 1 | SWE5N | rw | Integer32 | - | long int | Position register 5 (Cam) |
| 3636h | 1 | WPOS | ro | Integer8 | - | 0, 1, 2 | Enable Position Registers |
| 363Fh | 1 | WPOSE | ro | Integer32 | - | 0 ... 65535 | Enable Fast Position Registers 1 ... 16 |
| 3640h | 1 | WPOSP | rw | Integer32 | - | 0 ... 65535 | Polarity of Fast Position Registers 1 ... 16 |
| 3641h | 1 | WPOSX | rw | Integer32 | - | 0 ... 65535 | Mode of Fast Position Registers 1 ... 16 |
| 3643h | 1 | POSRSTAT | rw | Integer32 | - | - | Status of Fast Position Registers 1 ... 16 |
| 3644h | 1 | P1 | rw | Integer8 | - | 0 ... 3 | P1 Fast Position Register |
| 3645h | 1 | P2 | rw | Integer32 | - | long int | P2 Fast Position Register |
| 3646h | 1 | P3 | rw | Integer32 | - | long int | P3 Fast Position Register |
| 3647h | 1 | P4 | rw | Integer32 | - | long int | P4 Fast Position Register |
| 3648h | 1 | P5 | rw | Integer32 | - | long int | P5 Fast Position Register |
| 3649h | 1 | P6 | rw | Integer32 | - | long int | P6 Fast Position Register |
| 364Ah | 1 | P7 | rw | Integer32 | - | long int | P7 Fast Position Register |
| 364Bh | 1 | P8 | rw | Integer32 | - | long int | P8 Fast Position Register |
| 364Ch | 1 | P9 | rw | Integer32 | - | long int | P9 Fast Position Register |
| 364Dh | 1 | P10 | rw | Integer32 | - | long int | P10 Fast Position Register |
| 364Eh | 1 | P11 | rw | Integer32 | - | long int | P11 Fast Position Register |
| 364Fh | 1 | P12 | rw | Integer32 | - | long int | P12 Fast Position Register |
| 3650h | 1 | P13 | rw | Integer32 | - | long int | P13 Fast Position Register |
| 3651h | 1 | P14 | rw | Integer32 | - | long int | P14 Fast Position Register |
| 3652h | 1 | P15 | rw | Integer32 | - | long int | P15 Fast Position Register |
| 3653h | 1 | P16 | rw | Integer32 | - | long int | P16 Fast Position Register |

# Modes of operation

## Homing (DSP 402)

This mode enables to reference axis position measure. The position feedback is set to a value to express the position into machine mechanic physical dimensions.




| Lex 15 MP/HP | | Lex 15 LP | | | | |
|---|---|---|---|---|---|---|
| **Format** | **PDO mapping** | **Format** | **PDO mapping** | **Index** | **Sub** | **Name** |
| | | | | | | **HOMING MODE** |
| INTEGER32 | Yes | INTEGER32 | | 607Ch | 0 | Home offset |
| INTEGER8 | Yes | INTEGER8 | | 6098h | 0 | Homing method |
| INTEGER32 | Yes | UNSIGNED32 | | 6099h | 1 | Speed during search for switch |
| | | UNSIGNED32 | | 6099h | 2 | Speed during search for zero |
| UNSIGNED32 | Yes | UNSIGNED32 | | 609Ah | 0 | Homing acceleration |

The following table describes supported homing methods

## Lexium 15 LP

| Method as per DSP 402 | Brief description: Homing | ASCII command |
|---|---|---|
| - 128 ... -4 | Reserved | — |
| - 3 | Move to mechanical stop, with zeroing | NREF = 7 |
| - 2 | Set reference point at present position, allowing for lag/following error | NREF = 6 |
| - 1 | Homing within a single turn (direction of rotation depends on distance) | NREF = 5, DREF= 2 |
| 0 | Reserved | — |
| 1 | Homing to negative limit switch, with zeroing, negative count direction | NREF = 2, DREF= 0 |
| 2 | Homing to positive limit switch, with zeroing, positive count direction | NREF = 2, DREF= 1 |
| 3 ... 7 | Not supported | — |
| 8 | Homing to reference switch, with zeroing, positive count direction | NREF = 1, DREF= 1 |
| 9 ... 11 | Not supported | — |
| 12 | Homing to reference switch, with zeroing, negative count direction | NREF = 1, DREF= 0 |
| 13 ... 14 | Not supported | — |
| 15 ... 16 | Reserved | — |
| 17 | Homing to negative limit switch, without zeroing, negative count direction | NREF = 4, DREF= 0 |
| 18 | Homing to negative limit switch, without zeroing, positive count direction | NREF = 4, DREF= 1 |
| 19 ... 23 | Not supported | — |
| 24 | Homing to reference switch, without zeroing, positive count direction | NREF = 3, DREF= 1 |
| 25 ... 27 | Not supported | — |
| 28 | Homing to reference switch, without zeroing, negative count direction | NREF = 3, DREF= 0 |
| 29 ... 30 | Not supported | — |
| 31 ... 32 | Reserved | — |
| 33 | Homing within a single turn, negative count direction | NREF = 5, DREF= 0 |
| 34 | Homing within a single turn, positive count direction | NREF = 5, DREF= 1 |
| 35 | Set reference point at present position | NREF = 0 |
| 36 ... 127 | Reserved | — |

The following table describes supported homing methods

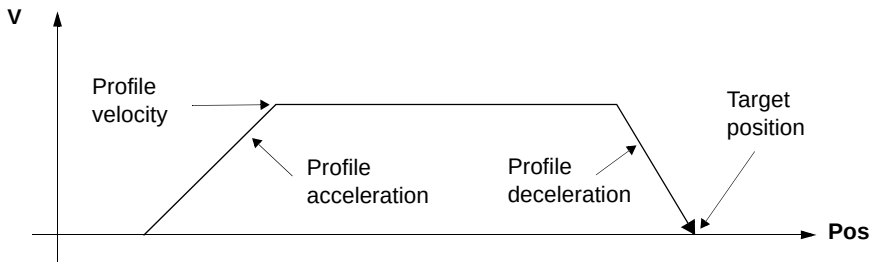
## Lexium 15 MP/HP

| Method as per DSP 402 | Brief description: Homing | ASCII command |
|---|---|---|
| - 128 ... -4 | Reserved | — |
| - 3 | Move to mechanical stop, with zeroing | NREF = 7 |
| - 2 | Set reference point at present position, allowing for lag/following error | NREF = 6 |
| - 1 | Homing within a single turn (direction of rotation depends on distance) | NREF = 5, DREF= 2 |
| 0 | Reserved | — |
| 1 | Homing to negative limit switch, with zeroing, negative direction of motion | NREF = 2, DREF= 0 |
| 2 | Homing to positive limit switch, with zeroing, positive direction of motion | NREF = 2, DREF= 1 |
| 3 ... 7 | Not supported | — |
| 8 | Homing to reference switch, with zeroing, positive direction of motion | NREF = 1, DREF= 1 |
| 9 ... 11 | Not supported | — |
| 12 | Homing to reference switch, with zeroing, negative direction of motion | NREF = 1, DREF= 0 |
| 13 ... 14 | Not supported | — |
| 15 ... 16 | Reserved | — |
| 17 | Homing to negative limit switch, without zeroing, negative direction of motion | NREF = 4, DREF= 0 |
| 18 | Homing to negative limit switch, without zeroing, positive direction of motion | NREF = 4, DREF= 1 |
| 19 ... 23 | Not supported | — |
| 24 | Homing to reference switch, without zeroing, positive direction of motion | NREF = 3, DREF= 1 |
| 25 ... 27 | Not supported | — |
| 28 | Homing to reference switch, without zeroing, negative direction of motion | NREF = 3, DREF= 0 |
| 29 ... 30 | Not supported | — |
| 31 ... 32 | Reserved | — |
| 33 | Homing within a single turn negative direction of rotation | NREF = 5, DREF= 0 |
| 34 | Homing within a single turn positive direction of rotation | NREF = 5, DREF= 1 |
| 35 | Set reference point at present position | NREF = 0 |
| 36 ... 127 | Reserved | — |

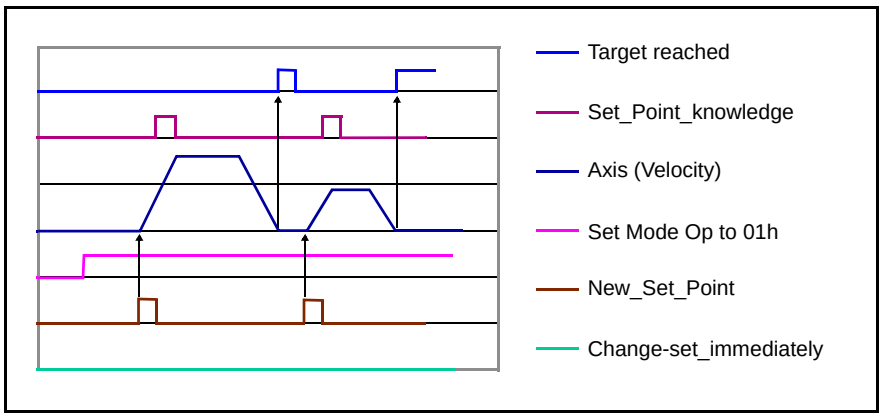
## Position profile (Point to point) (DSP 402)

This operation mode allows to do point to point position move control by a field bus. There are two ways to send a move sequence using the Change_set_immediately parameter:

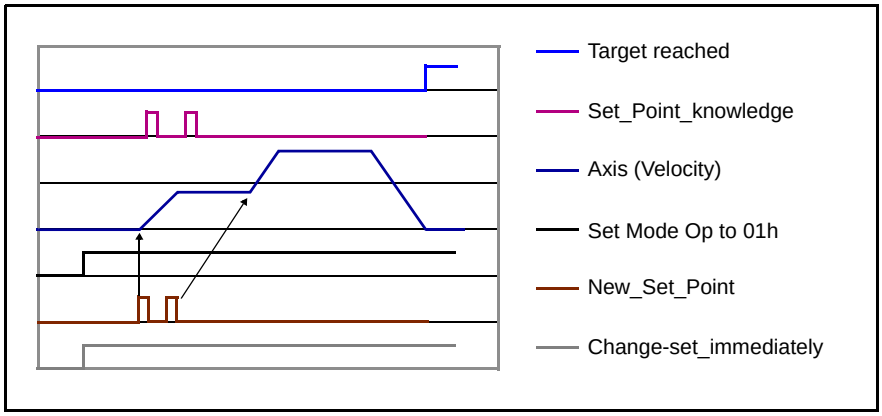
- By single set point: send set point, wait axis stop to send a new set point.
- By set of set points: send set point and send a new one when servo drive is able to receive a new one. Waiting axis stop is not compulsory.




### By single set point



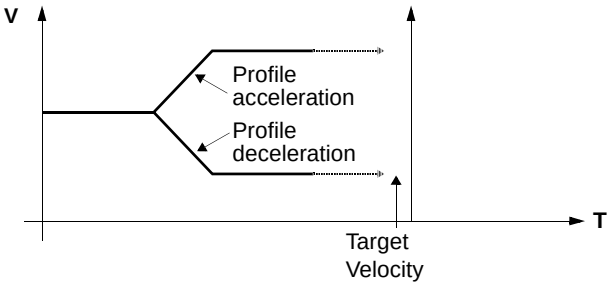

### By set of set points

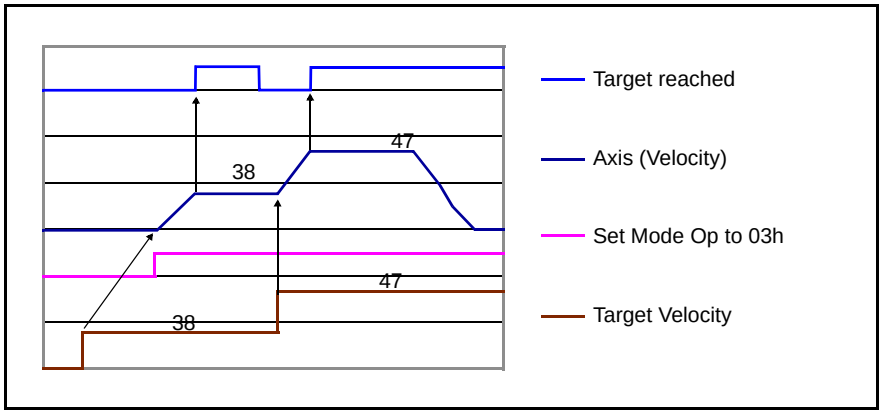


| Lex 15 MP/HP | | | Lex 15 LP | | | | | |
|---|---|---|---|---|---|---|---|---|
| Status | Format | PDO mapping | Status | Format | PDO mapping | Index | Sub | Name |
| | | | | | | | | |
| | INTEGER32 | Yes | | INTEGER32 | | 607Ah | 0 | Target position |
| | UNSIGNED8 | | -- | | | 607Bh | 0 | Position range limit |
| | INTEGER32 | | -- | | | 607Bh | 1 | Min position range limit |
| | INTEGER32 | | -- | | | 607Bh | 2 | Max position range limit |
| -- | | | | UNSIGNED32 | | 607Fh | 0 | Max profile velocity |
| -- | | | | UNSIGNED32 | | 6080h | 0 | Max motor speed |
| | UINTEGER32 | Yes | | UNSIGNED32 | Yes | 6081h | 0 | Profile velocity |
| | UNSIGNED32 | Yes | | UNSIGNED32 | Yes | 6083h | 0 | Profile acceleration |
| | UNSIGNED32 | Yes | | UNSIGNED32 | Yes | 6084h | 0 | Profile deceleration |
| -- | | | | UNSIGNED32 | | 6085h | 0 | Quick stop deceleration |
| | INTEGER16 | Yes | | INTEGER16 | Yes | 6086h | 0 | Motion profile type |
| -- | | | | UNSIGNED32 | | 60C5h | 0 | Max acceleration |
| -- | | | -- | | | 60C6h | 0 | Max deceleration |
| | | | | | | | | |
| | | | | | | | | **Motion profile types (accel/decel)** |
| Yes | | | Yes | | | |  | Trapeze |
| Yes | | | Yes | | | |  | Sinus² |
| Yes | | | Yes | | | | | Manufacturer profiles |

## Velocity profile (DSP 402)

Velocity profile is an operation mode that enables to assign new velocity set points which will be reached after acceleration or deceleration, which means it will not be directly applied to velocity servo loop.

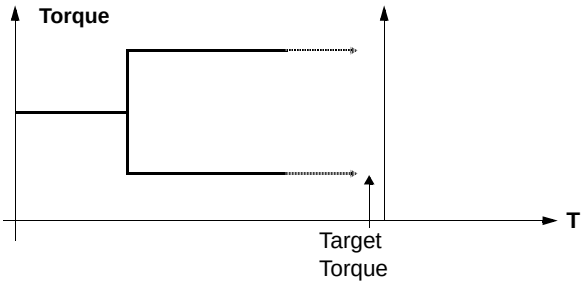


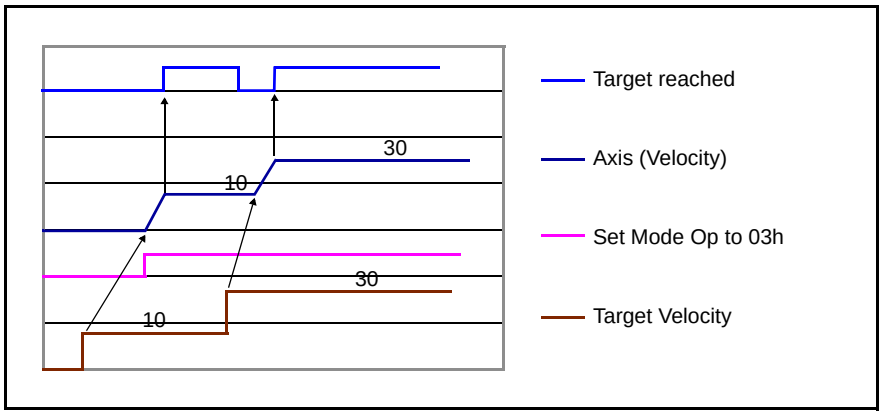




| Lexium 15 MP/HP | | Lexium 15 LP | | | | |
|---|---|---|---|---|---|---|
| Format | PDO mapping | Format | PDO mapping | Index | Sub | Name |
| | | **PROFILE VELOCITY MODE** | | | | |
| INTEGER32 | Yes | INTEGER32 | Yes | 606Ch | 0 | Velocity actual value |
| INTEGER32 | Yes | INTEGER32 | Yes | 60FFh | 0 | Target velocity |

## Profile torque (DSP 402)

The following diagram shows the profile torque specifications




| Lexium 15 LP | | | | |
|---|---|---|---|---|
| Format | PDO mapping | Index | Sub | Name |
| | | | | **PROFILE TORQUE MODE** |
| INTEGER16 | Yes | 6071h | 0 | Target torque |
| INTEGER16 | Yes | 6077h | 0 | Torque actual value |

## Position Controller Data

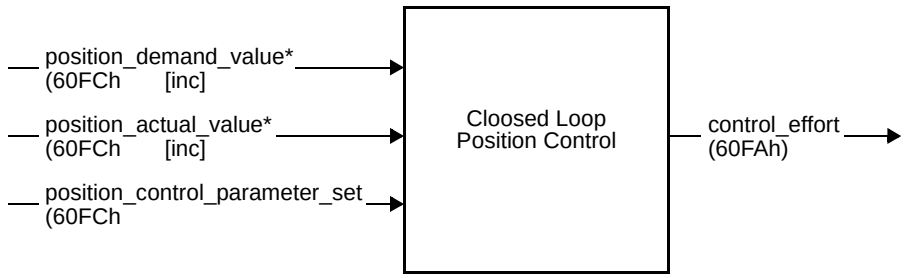
The objects below are the parameters used by the Lexium 15 Position Controller:

### Lexium 15 LP/ MP/HP

| Index | Sub index | ASCII | ACC ess | Format | Unit | Limits | Short description |
|---|---|---|---|---|---|---|---|
| 35C7h | 1 | PEMAX | rw | INTEGER32 | µm | long int | Max. following error |
| 35CAh | 1 | PGEARI | rw | INTEGER32 | µm | long int | Position resolution (Numerator) |
| 35CBh | 1 | PGEARO | rw | INTEGER32 | µm | long int | Position resolution (Denominator) |
| 35CFh | 1 | POSCNFG | rw | INTEGER8 | - | 0,1,2 | Axes type |

## Position Control (DSP 402)

The Position Control is not a Mode of Operation. It is a data setting to manage the Poition Loop of the servo drive:




| Lexium 15 MP/HP | | Lexium 15 LP | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Format | PDO mapping | Format | PDO mapping | Index | Sub | Name | Type | Attr. |
| | | | | | | **POSITION CONTROL FUNCTION** | | |
| INTEGER32 | Yes | INTEGER32 | Yes | 6063h | 0 | Position actual value* | INTEGER32 | ro |
| INTEGER32 | Yes | INTEGER32 | Yes | 6064h | 0 | Position actual value | INTEGER32 | ro |
| | | UNSIGNED32 | | 6065h | 0 | Following error window | UNSIGNED32 | rw |
| | | UNSIGNED32 | | 6067h | 0 | Position window | UNSIGNED32 | rw |
| | | UNSIGNED16 | | 6068h | 0 | Position window time | UNSIGNED16 | rw |

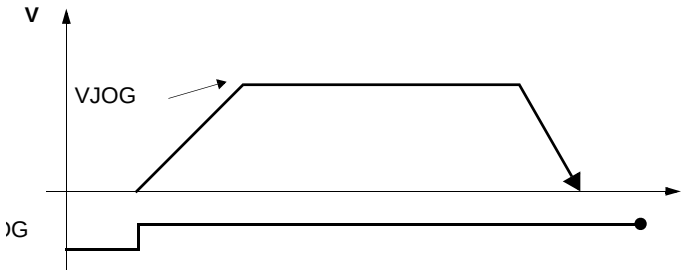
### PLEASE NOTE :

Ensure that BUSP7 ASCII command is set to 1. Another value of this ASCII will cause wrong value display in the objects 6063h and 6064h.

## Jog (only for Lexium 15 MP/HP)

Jog is a manufacturer Mode Of Operation:

***OPMODE F8h:*** ***Jog***

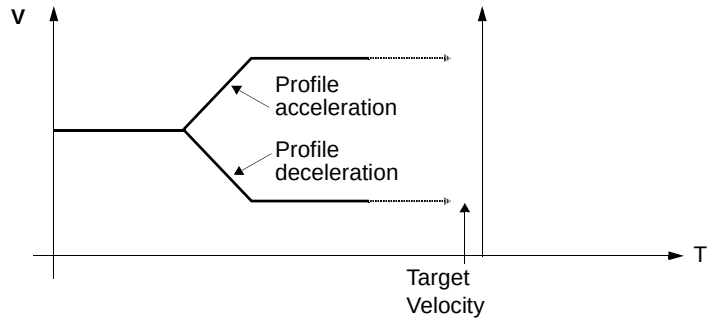



Lexium 15 MP/HP

| Index | Sub Index | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 3502h | 1 | ACCR | rw | Integer16 | >> ACCUNIT | 1 ... 32767 | Acceleration Ramp for homing/jog modes |
| 3524h | 1 | DECR | rw | Integer16 | >> ACCUNIT | 1 ... 32767 | Deceleration Ramp for homing/jog modes |
| 3591h | 1 | MJOG | Cmd | - | - | - | Start Jog Mode |
| 3621h | 1 | VJOG | rw | Integer32 | see VUNIT | 0 ... VLIM/VLIMN | Speed for Jog Mode |

## Digital Velocity (only for Lexium 15 MP/HP)

The Digital Velocity manufacturer Mode Of Operation is only available for Lexium 15 MP/HP. For Lexium 15 LP the DSP 402 Velocity Profile provides the same function.
The following diagram shows the acceleration and deceleration profile




The Acceleration and Deceleration profile are given in the objects used by Position Profile and Velocity Profile modes.

The Set Point is given by the following object:

| Index | Sub Index | Object | Description | Data Format | Attr |
|---|---|---|---|---|---|
| 2060h | 0 | VAR | Speed or current setpoint | Integer32 | r/w |

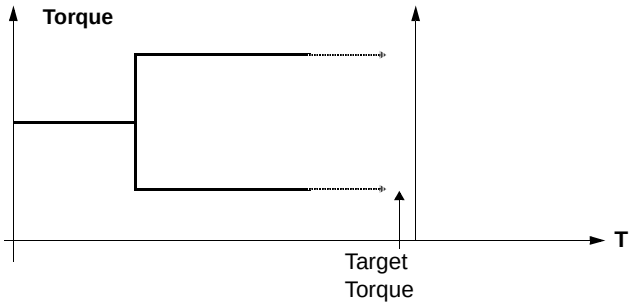
The value of the speed is given by the formula:

$$\text{Speed:} \qquad n[\text{min}^{-1}] = 1875 / 262144 \times \text{digital speed setpoint}$$

A new setpoint only becomes valid after Enable Operation has been refreshed (via SDO 6040h, Control word)

## Digital torque (only for Lexium 15 MP/HP)

The Digital Torque manufacturer Mode Of Operation is only available for Lexium 15 MP/HP. For Lexium 15 LP, the DSP 402 Torque Profile provides the same function.




The Set Point is given by the following object:

| Index | Sub Index | Object | Description | Data Format | Attr |
|---|---|---|---|---|---|
| 2060h | 0 | VAR | Speed or current setpoint | Integer32 | r/w |

The value is given by the formula:

$$\text{Current:} \qquad I[A] = \text{digital current setpoint} / 1640 \times I_{max}$$

A new set point only becomes valid after Enable Operation has been refreshed (via SDO 6040h, Control word)

## Gear manufacturer mode

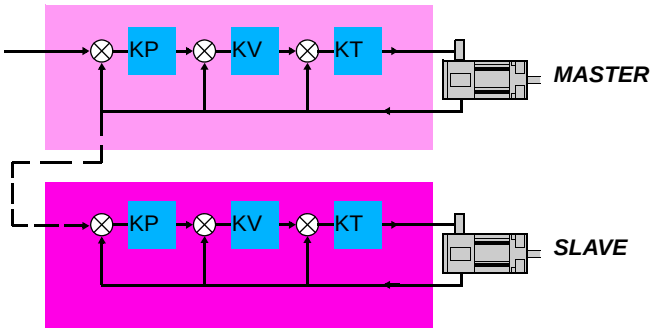
The manufacturer Mode of Operation enables electronic gearing. With CANopen it is possible to select the mode by setting F7h in Mode of Operation object (6060h).

The following diagram describes electronic gearing

***OPMODE 4:*** ***ELECTRONIC GEARING***

***The Master setpoint is given by the servo drive or the fieldbus***

***The Slave setpoint is given by the master servo drive***




However it is possible to modify the gear ratio parameters:

### Lexium 15 LP

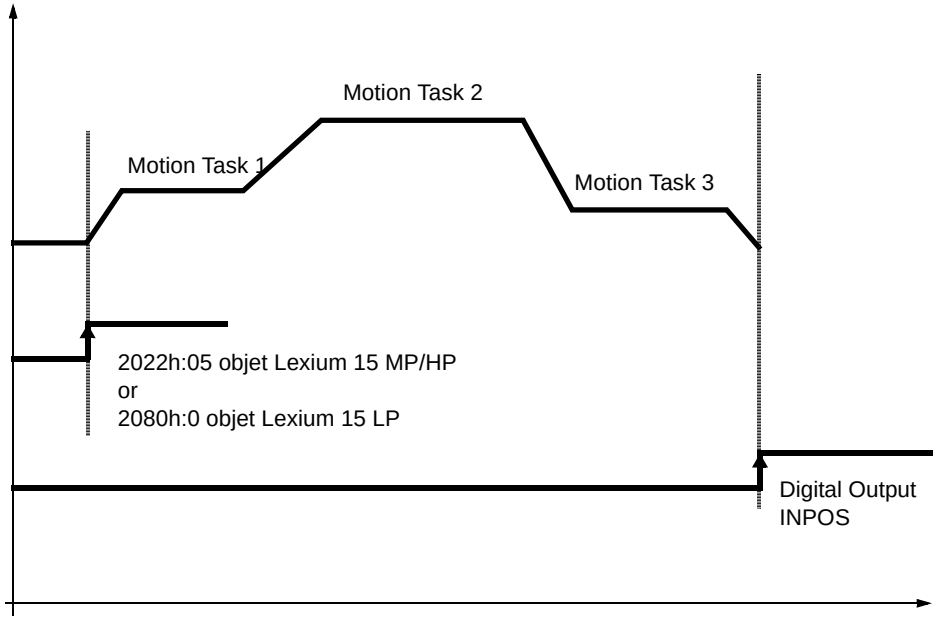
| Index | Sub Index | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 353Eh | 1 | GEARI | rw | Integer16 | - | 1 ... 32 767 | Input Factor for Electronic Gearing |
| 353Fh | 1 | GEARMODE | rw | Integer8 | - | 0 ... 17 | Position Input Electronic Gearing Mode |
| 3540h | 1 | GEARO | rw | Integer16 | - | -32 767 ... 32 767 | Output Factor for Electronic Gearing |
| 3656h | 1 | ROFFSABS | rw | Integer32 | PUNIT | long int | Reference Offset |

### Lexium 15 MP/HP

| Index | Sub Index | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 3532h | 1 | ENCIN | rw | Integer32 | Pulse/ Umdr. | 256,512, ... ,65 536 | Encoder Pulse Input |
| 353Eh | 1 | GEARI | rw | Integer16 | - | 1 ... 32767 | Input Factor for Electronic Gearing |
| 353Fh | 1 | GEARMODE | rw | Integer8 | - | 0 ... 17 | Secondary position source ( electr. Gearing or Secondary feedback to position controller) |
| 3540h | 1 | GEARO | rw | Integer16 | - | -32767 ... 32767 | Output Factor for Electronic Gearing |
| 3656h | 1 | ROFFS2 | rw | Integer32 | PUNIT | long int | Position offset for "absolute Gearing" |

## Motion Task

Motion Task may also be managed by CANopen. It enables to control the start of a specific Motion Task and to monitor its execution.
The following diagram shows a Motion Task sequence.




| Lexium 15 MP/HP | | | | Lexium 15 LP | | | | |
|---|---|---|---|---|---|---|---|---|
| Index | Sub index | Description | PDO mapping | Index | Sub index | Description | Format | PDO mapping |
| 2022h | 5 | Motion task number | Yes | 2080h | | Motion task number | UNSIGNED16 | Yes |
| | | | | 2081h | | Active motion task display | UNSIGNED16 | Yes |
| 1002h | 0 | Manufacturer status bit 19 InPosition | No | 1002h | | Manufacturer status bit 3 InPosition | UNSIGNED32 | Yes |
| 2070h | 11 | Enhanced status bit 19 InPosition | Yes | | | | | |

## Lexium 15 LP

The object 2080 sub index 0 indicates the number of the selected motion task.
Motion tasks 1 to 200 are EEPROM motion blocks, and motion tasks 201 to 300 are RAM motion blocks. The RAM motion blocks are loaded with the first 64 EEPROM motion blocks when the servo drive is switched on or reset. The object may be mapped in PDO.

The 2081 object shows the last motion task which has been set in the servo drive. If there is no value in object 2080h and if a motion task is started via the new-set point/set point acknowledge mechanism of the profile position mode, motion task 0 will be used and displayed. If you start a set of stored motion tasks (bit 3 of the motion task Control wordControl word O_C, SDO 35B9 sub1 set), the active motion task will be displayed in this object.

| Index | Sub Idx | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 351Ah | 1 | CLRORDER | Cmd | Integer16 | - | 0;1...180; 192...255 | Deleting a Motion Task |
| 351Dh | 1 | CONTINUE | Cmd | - | - | - | Continue last position order |
| 356Dh | 1 | INPOS | ro | Integer8 | - | - | Status of In-Position Signal |
| 35B6h | 1 | OVRIDE | rw | Integer8 | - | 0 ... 3 | Override Function for Motion Tasks |
| 35B7h | 1 | O_ACC | rw | Integer16 | | 0 ... 511 | Acceleration Time 1 for Motion Task 0 |
| 35B9h | 1 | O_C | rw | Integer16 | - | int | Control for Motion Task 0 |
| 35BAh | 1 | O_DEC | rw | Integer16 | - | int | Braking Time 1 for Motion Task 0 |
| 35BCh | 1 | O_FN | rw | Integer16 | - | 0, 1, ... ,180,192 ... 255 | Next Task Number for Motion Task 0 |
| 35BDh | 1 | O_FT | rw | Integer16 | Milliseconds | 1 ... 32 767 | Delay before Next Motion Task |
| 35BEh | 1 | O_P | rw | Integer32 | - | long int | Target Position/Path for Motion Task 0 |
| 35BFh | 1 | O_V | rw | Integer32 | - | long int | Target Speed for Motion Task 0 |
| 35D6h | 1 | PTMIN | rw | Integer16 | >>ACCUNIT | 1 ... 32 767 | Min. Acceleration Ramp for Motion Tasks |
| 35D9h | 1 | PVMAXN | rw | Integer32 | VUNIT | 0 ... VLIM | Max. (Negative) Velocity for Position Control |
| 3642h | 1 | MOVE | Cmd | Integer16 | - | 0,1, ... ,180,192 ... 255 | Command Start Motion Task |
| 3654h | 1 | PTARGET | rw | Integer32 | - | long int | Last Target Position |
| 3659h | 1 | ACCUNIT | rw | Integer32 | - | 0, 1, ... , 5 | Type of acceleration setpoint for the system |
| 365Bh | 1 | MTMUX | rw | Integer16 | - | 0, 192 ... 255 | Presetting for motion task that is processed later |
| 36B6h | 1 | DOVRIDE | rw | Integer16 | - | 0 ... 8 192 | Digital Override Factor |
| 36D2h | 1 | NREFMT | rw | Integer16 | | 0 ... 511 | Homing with following motion task |

## Lexium 15 MP/HP

The object 2022 sub index 05 indicates the number of the selected motion task.
Motion tasks 1 to 180 are EEPROM motion blocks, and motion tasks 192 to 255 are RAM motion blocks. The RAM motion blocks are loaded with the first 64 EEPROM motion blocks when the servo drive is switched on or reset. The object may be mapped in PDO.

| Index | Sub Idx | ASCII | Access | Format | Units | Limits | Short decription |
|---|---|---|---|---|---|---|---|
| 351Ah | 1 | CLRORDER | Cmd | Integer16 | - | 0;1 ... 180; 192 ... 255 | Deleting a Motion Task |
| 351Dh | 1 | CONTINUE | Cmd | - | - | - | Continue last position order |
| 356Dh | 1 | INPOS | ro | Integer8 | - | - | Status of In-Position Signal |
| 35B6h | 1 | OVRIDE | rw | Integer8 | - | 0 ... 3 | Override Function for Motion Tasks |
| 35B7h | 1 | O_ACC1 | rw | Integer16 | $mm/s^2$ | 1 ... 32 000 | Acceleration Time 1 for Motion Task 0 |
| 35B8h | 1 | O_ACC2 | rw | Integer16 | ms | 1 ... 32 000 | Acceleration Time 2 for Motion Task 0 |
| 35B9h | 1 | O_C | rw | Integer16 | - | int | Control Variable for Motion Task 0 |
| 35BAh | 1 | O_DEC1 | rw | Integer16 | $mm/s^2$ | 1 ... 32 000 | Braking Time 1 for Motion Task 0 |
| 35BBh | 1 | O_DEC2 | rw | Integer16 | ms | 1 ... 32 000 | Deceleration Time 2 for Motion Task 0 |
| 35BCh | 1 | O_FN | rw | Integer16 | - | 0, 1, ... ,180,192 ... 255 | Next Task Number for Motion Task 0 |
| 35BDh | 1 | O_FT | rw | Integer16 | ms | 1 ... 32 767 | Delay before Next Motion Task |
| 35BEh | 1 | O_P | rw | Integer32 | - | long int | Target Position/Path for Motion Task 0 |
| 35BFh | 1 | O_V | rw | Integer32 | - | long int | Target Speed for Motion Task 0 |
| 35D6h | 1 | PTMIN | rw | Integer16 | >>ACCUNIT | 1 ... 32 767 | Min. Acceleration Ramp for Motion Tasks |
| 35D9h | 1 | PVMAXN | rw | Integer32 | VUNIT | 0 ... VLIM | Max. (Negative) Velocity for Position Control |
| 35F5h | 1 | SPSET | rw | Integer8 | - | 0, 1, 2, 3 | Enable for S-curve |
| 362Eh | 1 | IN2PM | rw | Integer8 | - | 0, 1, 2 | In-Position 2 Mode |
| 3642h | 1 | MOVE | Cmd | Integer16 | - | 0,1, ... ,180,192 ... 255 | Start Motion Task |
| 3654h | 1 | PTARGET | rw | Integer32 | - | long int | Last Target Position |
| 365Bh | 1 | MTMUX | rw | Integer16 | - | 0, 192 ... 255 | Presetting for motion task that is processed later |

# Manufacturer Data (Manufacturer status data)

| Lexium 15 MP/HP | | Lexium 15 LP | | | | |
|---|---|---|---|---|---|---|
| Format | PDO mapping | Format | PDO mapping | Index | Sub | Name |
| | | | | | | |
| UINTEGER32 | | UNSIGNED32 | | 1000h | 0 | Device type |
| UINTEGER8 | | UNSIGNED8 | | 1001h | 0 | Error register |
| UINTEGER32 | | UNSIGNED32 | Yes | 1002h | | Manufacturer-specific status register |

## Factor group

There is a possibility of conversion between physical dimensions and sizes, and the internal units used in the device (increments). Several factors can be implemented. This chapter describes how these factors influence the system, how they are calculated and which data are necessary to create them.
**Normalized parameters are denoted by an asterisk *.**

### Relationship between Physical and Internal Units

The factors defined in the factor group set up a relationship between device-internal units (increments) and physical units.
The factors are the result of the calculation of two parameters called dimension index and notation index. The dimension index indicates the physical dimension; the notation index indicates the physical unit and a decimal exponent for the values. These factors are directly used to normalize the physical values.

The notation index can be used in two ways:

- For a unit with decimal scaling and notation index < 64, the notation index defines the exponent/ decimal place of the unit.
- For a unit with non-decimal scaling and notation index > 64, the notation index defines the sub-index of the physical dimension of the unit.

| **Lexium 15 MP/HP** | | **Lexium 15 LP** | | | | |
|---|---|---|---|---|---|---|
| Format | PDO mapping | Format | PDO mapping | Index | Sub | Name |
| | | | | | | **FACTOR GROUP** |
| INTEGER8 | Yes | INTEGER8 | | 6089h | 0 | Position notation index |
| UNSIGNED8 | Yes | UNSIGNED8 | | 608Ah | 0 | Position dimension index |
| INTEGER8 | Yes | INTEGER8 | | 608Bh | 0 | Velocity notation index |
| UNSIGNED8 | Yes | UNSIGNED8 | | 608Ch | 0 | Velocity dimension index |
| INTEGER8 | Yes | INTEGER8 | | 608Dh | 0 | Acceleration notation index |
| UNSIGNED8 | Yes | UNSIGNED8 | | 608Eh | 0 | Acceleration dimension index |
| UNSIGNED8 | | UNSIGNED8 | | 6093h | 0 | Position factor |
| UNSIGNED32 | Yes | UNSIGNED32 | | 6093h | 1 | Numerator |
| UNSIGNED32 | Yes | UNSIGNED32 | | 6093h | 2 | Feed constant |
| | | UNSIGNED8 | | 6094h | 0 | Velocity encoder factor |
| UNSIGNED32 | Yes | UNSIGNED32 | | 6094h | 1 | Numerator |
| UNSIGNED32 | Yes | UNSIGNED32 | | 6094h | 2 | Divisor |
| UNSIGNED32 | Yes | UNSIGNED32 | | 6097h | 1 | Numerator |
| UNSIGNED32 | Yes | UNSIGNED32 | | 6097h | 2 | Divisor |

## Lexium 15 MP/HP

### Object 608Bh

This Object determines the notation for SDO 6081h profile_velocity. In combination with SDO 608Ch velocity_dimension_index, the following basic units can be represented:

| physical dimension | units | velocity dimension index | velocity notation index |
|---|---|---|---|
| manufacturer-specific | incr/sec | 0 | 0 |
| revolutions | rpm | 11 | 73 |

### Object 608Ch

This Object determines the dimension for SDO 6081h profile_velocity. In combination with SDO 608Bh velocity_notation_index, the following basic units can be represented

| physical dimension | units | velocity dimension index | velocity notation index |
|---|---|---|---|
| manufacturer-specific | incr/sec | 0 | 0 |
| revolutions | rpm | 11 | 73 |

### Object 6093h

The postion factor converts the target position into the internal data format of the Lexium 15 MP/HP (increments).
The position controller can be run with the resolution set to 20 bits/turn or 16 bits/turn (see SDO 35D1h and the ASCII command PRBASE).
The numerator and the feed constant can be used to set up any value of scaling.

$$\text{position\_factor} = \frac{\text{position\_encoder\_resolution} \times \text{gear\_ratio}}{\text{feed\_constant}}$$

- Position_encoder_resolution
  resolution of the position controller is $2^{20}$ or $2^{16}$
- Gear_ratio
  transmission ratio for the gearing that is used
- feed_constant
  the feed constant of the output side of the servo drive gearing

Example: One turn is equivalent to 10 000 increments. The gear ratio is 1.

$$\text{position\_factor} = \frac{2^{20}\ \text{incr}}{1000\ \text{incr}}$$

- Numerator: $2^{20}$
- Feed constant 10000
- Setpoint provision in [incr/turn] for SDO 607Ah (target_position)

The Numerator corresponds to the ASCII-parameter PGEARO, feed-constant to the parameter PGEARI.

### Object 6094h

The velocity_encoder_factor converts the target speed (rpm) or velocity (incr/sec) into the terminal data format of the Lexium 15 MP/HP (increments). The following formula and associated table explain how the velocity_encoder_factor is calculated:

$$\text{velocity\_encoder\_factor} = \frac{\text{velocity\_encoder\_resolution} \times \text{gear\_ratio} \times \text{position\_unit} \times \text{Fvelocity(notation\_index)}}{\text{feed\_constant} \times \text{velocity\_unit} \times \text{sec} \times \text{Fposition(notation\_index)}}$$

| Parameter | Designation |
|---|---|
| Velocity_encoder_resolution | Resolution of the speed $2^{20}$ |
| Gear_ratio | Transmission ratio for the gearing that is used |
| Position_unit | In meters |
| Fposition (notation_index) | Dimension_index=1, notation_index=0[m] |
| Feed_constant | Feed constant for the output side of the servo drive gearing |
| Velocity_unit | In [m/s] |
| Fvelocity (notation_index) | Dimension_index=13, notation_index=0 [m/s] |

(see also SDO 606Bh velocity_notation_index and SDO 606Ch velocity_dimension_index)

Example: The speed-setpoint has to be made in rounds per minute (rpm/min).
The transmission ratio and feed constant are 1

$$\text{velocity\_encoder\_factor} = \frac{2^{20}}{1} \quad \frac{1/1}{1} \quad \frac{\text{incr}}{\text{s}}$$

Expanded for rpm:

$$\text{velocity\_encoder\_factor} = \frac{2^{20}}{1} \quad \frac{1/1}{1} \quad \frac{[\text{m}]}{[\text{m/s}]} \quad \frac{\text{incr}}{\text{s}} \times \frac{1}{60[\text{s/min}]} = \frac{2^{20}}{60} \quad \frac{\text{incr}}{\text{s}}$$

- Numerator: $2^{20}$
- Divisor: 60
- Setpoint provision in [rpm] for SDO 60FFh target_velocity/velocity units
- Setpoint provision in [incr/min] for SDO 6081h profile_velocity/speed units

Since the speed controller operates internally with a resolution of $2^{20}$ bits/turn, regardless of the resolution of the encoder system, the following expression is used to calculate the operating mode pv ( for revolutions per minute):

$$\text{increments} = \frac{262144}{1875} \times \text{speed setpoint}[\quad \text{min}^{-1}]$$

This incremental setpoint provison should be used for cyclic applications (for example, position control, 4 ms cycle). The advantages are there is no rounding error and a lower CPU loading.
The above calculation is valid if divisor or numerator is set to 0.
The velocity_encoder_factor also affects SDO 6081h profile_velocity. In order to be able to use this factor for the position mode (pp) as well, the internal gearing factors PGEARI and PGEARO must be equal (PGEARI=PGEARO; SDO 2020h sub-index; 08h, 09h). If the divisor or numerator is set to 0, then the internal scaling is used: increments per cycle (250 µs).

### Object 6097h

The acceleration_factor converts the acceleration [unit: $s^2$] into the internal format of the Lexium 15 MP/HP.
At present, the numerator and divisor are read-only. The values are set to 1. If the acceleration_factor is set to 1, then the ramp settings (SDO 6083h, profile_acceleration and SDO 6084h, profile_deceleration) will be provided as acceleration times [ms] required to reach the target speed (SDO 6081h profile_velocity).

## Lexium 15 LP

### Object 6089h

The "position notation index" scales position setpoints, which units are defined with the "position dimension index" in SI-units, in power of ten.
The following table describes the relationship between the values for SDO 6089 and the manufacturer specific parameter PUNIT:

| Value of SDO6089h | ASCII parameter PUNIT | Scaling |
|---|---|---|
| FFh | 1 | $10^{-1}$ |
| FEh | 2 | $10^{-2}$ |
| FDh | 3 | $10^{-3}$ |
| FCh | 4 | $10^{-4}$ |
| FBh | 5 | $10^{-5}$ |
| FAh | 6 | $10^{-6}$ |
| F9h | 7 | $10^{-7}$ |
| F8h | 8 | $10^{-8}$ |
| F7h | 9 | $10^{-9}$ |
| 0 | 0 | 1 |

### Object 608Ah

The "position dimension index" defines the SI-units of the used position setpoints.
The following table describes the relationship between the SDO values and the manufacturer specific parameter PUNIT:

| Value of SDO608Ah | ASCII parameter PUNIT | SI unit |
|---|---|---|
| 1 | 9...1 | m |
| 0 | 0 | Manufacturer specific increments |

The parameter PUNIT can be stored in the servo drive. The values of 6089h and 608Ah are initialized by that paremeter.

### Object 608Fh

The position encoder resolution defines the ratio of encoder increments per motor revolution. This object is used in the same way for SDO 6090 (velocity encoder resolution).

$$\text{position encoder resolution} = \frac{\text{encoder increments}}{\text{motor revolutions}}$$

### Object 6091h

The gear ratio defines the ratio of motor shaft revolution per driving shaft revolution. This includes the gear if present.

$$\text{gear ratio} = \frac{\text{motor shaft revolutions}}{\text{driving shaft revolutions}}$$

### Object 6092h

The feed constant defines the ratio of motor shaft revolution per driving shaft revolution. This includes the gear if present.

$$\text{feed constant} = \frac{\text{feed}}{\text{driving shaft revolutions}}$$

### Object 6093h

The position factor converts the desired position (in position units) into the internal format (in increments). These values are calculated via the SDOs 608F and 6091.

$$\text{position factor} = \frac{\text{position encoder resolution} \bullet \text{gear ratio}}{\text{feed constant}}$$

### Object 608Bh

The "velocity notation index" scales velocity setpoints, which units are defined with the "velocity dimension index" as SI-units, in power of ten.
The following table describes the relationship between the SDO values and the parameter VUNIT:

| Value of SDO608Bh | ASCII parameter VUNIT | Scaling |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 5 | 1 |
| 0 | 6 | 1 |
| FDh | 7 | $10^{-3}$ |
| FDh | 8 | $10^{-3}$ |

### Object 608Ch

The "velocity dimension index" defines the SI-unit of the used velocity setpoint.
The following table describes the relationship between the SDO values and the manufacturer specific parameter VUNIT:

| Value of SDO608Ch | ASCII parameter VUNIT | SI unit |
|---|---|---|
| A6h | 0 | m/s |
| A4h | 1 | round/min (rpm) |
| A6h | 5 | m/s |
| A7h | 6 | m/min |
| A6h | 7 | m/s |
| A7h | 8 | m/min |

The parameter VUNIT can be stored in the servo drive. The values for SDOs 608Bh ad 608Ch are initialized by this parameter. Only the described values for VUNIT are possible with the profile DS 402.

### Object 608Dh

The "acceleration notation index" scales acceleration setpoints, which units are defined with the "acceleration dimension index" as SI-units, in power of ten.
The following table describes the relationship between the SDO values and the parameter ACCUNIT:

| Value of SDO608Dh | ASCII parameter ACCUNIT | Scaling |
|---|---|---|
| 0 | 1,5 | 1 |
| FAh | 3 | $10^{-6}$ |
| FDh | 4 | $10^{-3}$ |

### Object 608Eh

The "acceleration dimension index" defines the SI-units of the used acceleration setpoints.
The following table describes the relationship between the SDO values and the manufacturer specific parameter ACCUNIT:

| Value of SDO608Eh | ASCII parameter ACCUNIT | SI unit |
|---|---|---|
| AEh | 1 | $rad/s^2$ |
| 55h | 3, 4, 5 | m/s |

The parameter ACCUNIT can be stored in the servo drive. The values for SDOs 608Dh ad 608Eh are initialized by this parameter. Only the described values for ACCUNIT are possible with the profile DS 402.

### Object 6097h

The acceleration factor converts the acceleration (in acceleration units/s) into the internal format (in increments/s). This factor is actually calculated from SDO 6093 and readable only.

$$\text{acceleration factor} = \frac{\text{velocity unit} \bullet \text{encoded factor}}{\text{acceleration unit} \bullet \text{second}}$$

## Identification

The following tables list the Identification PDOs

| Format | PDO mapping | Index | Sub | Name |
|---|---|---|---|---|
| | | | | |
| UNSIGNED32 | | 1000h | 0 | Device type |
| UNSIGNED8 | | 1001h | 0 | Error register |
| UNSIGNED32 | Yes | 1002h | 0 | Manufacturer-specific status register |

| Index | Sub idx | Var Type | Description | Format | Access |
|---|---|---|---|---|---|
| 1008h | 0 | VAR | Manufacturer device name | Visible String | r |
| 100Ah | 0 | VAR | Manufacturer software version | Visible String | r |
| 1018h | | RECORD | Identity Object | | |
| 1018h | 0 | VAR | Number of entries | UINTEGER8 | r |
| 1018h | 1 | VAR | Vendor ID | UINTEGER32 | r |
| 1018h | 2 | VAR | Product Code | UINTEGER32 | r |
| 1018h | 3 | VAR | Revision number | UINTEGER32 | r |
| 1018h | 4 | VAR | Serial number | UINTEGER32 | r |

## Errors

The following tables list the different error codes (Emergency messages) for Lexium 15 LP and Lexium 15 MP/HP

### Lexium 15 MP/HP

| SDO | 1003h | DS301 |
|---|---|---|
| Error Code | Category | Description |
| 0000h | — | Error reset or no error (mandatory) |
| 1000h | — | Generic error (mandatory) |
| 1080h | 5 | No R1A/R1C (status not ready for operation) |
| 2330h | 2 | Earth short (F22) |
| 3100h | 1 | No mains/line-BTB (F16) |
| 3110h | 1 | Overvoltage in DC-bus/DC-link (F02) |
| 3120h | 1 | Undervoltage in DC-bus/DC-link (F05) |
| 3130h | 1 | Supply line phase missing (with PMODE = 2) (F19) |
| 4110h | 1 | Ambient temperature too high (F13) |
| 4210h | 1 | Heat sink temperature too high (F01) |
| 4310h | 1 | Motor temperature too high (F06) |
| 5111h | 1 | Fault in ⎓ 15 V external control power supply (F07) |
| 5380h | 1 | Fault in A/D converter (F17) |
| 5400h | 1 | Fault in output stage (F14) |
| 5420h | 1 | Braking (chopper) (F18) |
| 5441h | 1 | Operating error for PWR-option (F27) |
| 5530h | 1 | Serial EEPROM (F09) |
| 5581h | 1 | Flash EEPROM (F10) |
| 6010h | 4 | Watchdog (software reset, F32) |
| 6181h | 4 | BCC error (table) |
| 6182h | 4 | BCC error (system macro) |
| 6183h | 4 | BCC error (serial EEPROM) |
| 6184h | 4 | FPGA error |
| 6185h | 4 | Fault/error (table) |
| 6281h | 4 | User software BCC (macro, F32) |

| SDO | 1003h | DS301 |
|---|---|---|
| Error Code | Category | Description |
| 6282h | 4 | Faulty user software (macro, F32) |
| 6320h | 3 | Parameter error |
| 7111h | 1 | Braking error/fault (F11) |
| 7122h | 1 | Commutation error (F25) |
| 7181h | 5 | Could not enable Lexium |
| 7182h | 5 | Command only possible in disabled status |
| 7303h | 1 | Feedback device error (F04) |
| 8053h | 1 | Handling error (F21) |
| 8181h | 2 | Response monitoring activated |
| 8182h | 1 | CAN bus off (F23) |
| 8281h | 5 | Status machine not in operation enable condition |
| 8282h | 5 | Wrong mode setting |
| 8331h | 1 | I2t (torque fault, F15) |
| 8480h | 1 | Overspeed (F08) |
| 8611h | 2 | Lag/following error |
| 8681h | 5 | Invalid motion task number |
| 8682h | 2 | External trajectory error (F28) (only with Sercos) |
| FF01h | 4 | Serious exception error (F32) |
| FF02h | 3 | Error in PDO elements |
| FF03h | 5 | Operating mode |
| FF04h | 1 | Slot error (F20) |
| FF06h | 2 | Warning display as error (F24) |
| FF07h | 2 | Homing error (drove onto HW limit switch) (F26) |
| FF08h | 2 | Sercos error (F29) |
| FF11h | 1 | Sercos |

## Lexium 15 LP

| SDO | 1003h | DS301 |
|---|---|---|
| **Error Code** | **Category** | **Description** |
| 0000h | — | Error reset or no error (mandatory) |
| 1000h | — | Generic error (mandatory) |
| 1080h | 5 | No R1A/R1C (status not ready for operation) |
| 2330h | 1 | Error in ground connection (F22) |
| 2380h | 1 | Error in motor connection (phase fault) (F12) |
| 3100h | 1 | No mains/line-BTB (F16) |
| 3110h | 1 | Overvoltage in DC-bus/DC-link (F02) |
| 3120h | 1 | Undervoltage in DC-bus/DC-link (F05) |
| 3130h | 1 | Supply line phase missing (with PMODE = 2) (F19) |
| 4110h | 1 | Ambient temperature too high (F13) |
| 4210h | 1 | Heat sink temperature too high (F01) |
| 4310h | 1 | Motor temperature too high (F06) |
| 5111h | 1 | Fault in ⎓ 15 V auxiliary voltage (F07) |
| 5380h | 1 | Fault in A/D converter (F17) |
| 5400h | 1 | Fault in output stage (F14) |
| 5420h | 1 | Ballast (chopper) (F18) |
| 5441h | 1 | Operating error for PWR-option (F27) |
| 5530h | 1 | Serial EEPROM (F09) |
| 6320h | 3 | Parameter error |
| 7111h | 1 | Braking error/fault (F11) |
| 7122h | 1 | Commutation error (F25) |
| 7181h | 5 | Could not enable Lexium |
| 7303h | 1 | Feedback device error (F04) |
| 7305h | 1 | Signal failure digital encoder input (F10) |
| 8053h | 1 | Handling error (F21) |
| 8182h | 1 | CAN bus off (F23) |
| 8331h | 1 | I2t (torque fault, F15) |
| 8480h | 1 | Overspeed (F08) |
| 8611h | 2 | Lag/following error (n03/F03) |
| 8681h | 5 | Invalid motion task number |
| FF01h | 4 | Serious exception error (F32) |
| FF02h | 3 | Error in PDO elements |
| FF04h | 1 | Slot error (F20) |
| FF06h | 2 | Warning display as error (F24) |
| FF07h | 2 | Homing error (drove onto HW limit switch) (F26) |
| FF08h | 2 | Sercos error (F29) |
| FF11h | 2 | Emergency timeout failure(F30) |

## Warnings/ Manufacturer Status Register

The following tables list the different warnings for Lexium 15 LP and Lexium 15 MP/HP

| SDO | | 1002h | | | |
|---|---|---|---|---|---|
| Lexium 15 MP/HP | | | Lexium 15 LP | | |
| Bit | | Explanation | Bit | | Explanation |
| 0 | n01 | Warning 1: I²t-threshold exceeded | 0 | n01 | L²t threshold exceeded |
| 1 | n02 | Warning 2: full braking power reached | 1 | n02 | Reached preset braking power limit |
| 2 | n03* | Warning 3: lag/following error | 2 | n03* | Exceeded preset contouring error limit |
| 3 | n04* | Warning 4: response monitoring activated | 3 | n04* | Nodeguarding monitoring has been activated |
| 4 | n05 | Warning 5: power supply phase missing | 4 | n05 | Power supply phase missing |
| 5 | n06* | Warning 6: software limit-switch 1 has been activated | 5 | n06* | Position fall below software limit switch 1 |
| 6 | n07* | Warning 7: software limit-switch 2 has been activated | 6 | n07* | Position exceeded software limit switch 2 |
| 7 | n08 | Warning 8: faulty motion task started | 7 | n08 | A faulty motion task was started |
| 8 | n09 | Warning 9: no reference point set at start of motion task | 8 | n09 | No reference point (Home) set at start of motion task |
| 9 | n10* | Warning 10: PSTOP activated | 9 | n10* | PSTOP limit-switch activated |
| 10 | n11* | Warning 11: NSTOP activated | 10 | n11* | NSTOP limit-switch activated |
| 11 | n12 | Warning 12: motor default values were loaded | 11 | n12 | Only for ENDAT or HIPERFACE® : discrepancy between servo motor number saved in the encoder and the servo motor default values loaded |
| 12 | n13* | Warning 13: expansion board not functioning correctly | 12 | n13* | Expansion card not operating correctly |
| 13 | n14 | Warning 14: motor phase | 13 | n14 | SinCos commutation (wake & shake) not completed, will be canceled when servo drive is enabled and wake & shake carried out |
| 14 | n15 | Warning 15: erroneous VCT entry selected | 14 | n15 | Fault according to speed/current table INXMODE 35 |
| 15 | n16 | Warning 16: reserve | 15 | n16 | Summarized warning for n17 to n31 |
| 16 | | Motion task active | 16 | n17 | CAN-Sync is not logged in (with SYNCSRC=3) |
| 17 | | Reference point set | 17 | n18 | Max. number of turns exceeded |
| 18 | | Actual position = home position | 18 | n19 | Reserved |
| 19 | | In Position | 19 | n20 | Reserved |
| 20 | | Position latch made (positive edge) | 20 | n21 | Reserved |
| 21 | | Reached Position 0 | 21 | n22 | Reserved |
| 22 | | Reached Position 1 | 22 | n23 | Reserved |
| 23 | | Reached Position 2 | 23 | n24 | Reserved |
| 24 | | Reached Position 3 | 24 | n25 | Reserved |
| 25 | | Reached Position 4 | 25 | n26 | Reserved |
| 26 | | Finished initialization | 26 | n27 | Reserved |
| 27 | | Reached Position 5 | 27 | n28 | Reserved |
| 28 | | Speed = 0 | 28 | n29 | Reserved |
| 29 | | Safety relay has been activated | 29 | n30 | Reserved |
| 30 | | Output stage enabled | 30 | n31 | Reserved |
| 31 | | Error present | 31 | n32 | Firmware is an unreleased beta version |

*** = these warning messages result in a controlled shut-down of the servo drive (braking by emergency stop ramp)**

# Lexium 15 MP/HP

## Response to BUSOFF communication faults

The communication fault BUSOFF is directly monitored and signaled by Level 2 (CAN controller). This message may have various causes. The list below provides a few examples:

- telegrams are transmitted, although there is no other CAN node connected,
- CAN nodes have different transmission rates,
- the bus cable is faulty,
- faulty cable termination causes reflections on the cable.

A BUSOFF is signaled by the Lexium 15 MP/HP only if another CAN node is connected and if at least one Object was successfully transmitted. The BUSOFF condition is signaled by the error message F23. If the output stage is enabled for the execution of a motion task when this fault occurs, then the servo drive is braked to a stop, using the emergency stop ramp. The output stage is thus disabled.

## Setting the station address

Three different ways enable to configure the esalve address (address on the CAN-Bus) for the servo drive:

- using the pushbuttons on the front panel (see installation instructions for Lexium 15),
- with the commissioning software Unilink L or Unilink MP,
- using the ASCII command sequence: ADDR nn  SAVE => COLDSTART (nn = address).

The address range can be expanded from 1 ... 63 to 1 ... 127 by using the ASCII-Object MDRV.

**⚠ CAUTION**

**RISK OF UNEXPECTED OPERATION**

After changing the station adress and/or the baud rate you must turn the ⎓ 24 V external control power supply for the servo drive off and on again.

**Failure to follow this instruction can result in injury or equipment damage.**

## Setting the transmission rate

Three different ways enable to configure the CAN transmission rate:

- using the pushbuttons on the front panel (see installation instructions for Lexium 15 servo drive),
- with the commissioning software Unilink L or Unilink MP,
- using the ASCII command sequence: CBAUD bb => SAVE => COLDSTART (bb = transmission rate in kbit/s).

The following table shows the coding of the transmission rate in the LED display:

| Transmission rate: kbit/s | Max. cable length: m |
|---|---|
| 2 | 20 |
| 5 | 50 |
| 10 | 100 |
| 12 | 125 |
| 25 | 250 |
| 33 | 333 |
| 50 | 500 (Default) |
| 66 | 666 |
| 80 | 800 |
| 100 | 1000 |

# Basic data types

## Unsigned integer

Data in the basic data type UNSIGNEDn define exclusively positive integers.
The value range is from 0 to $2^{n-1}$. The bit sequence b = $b_{n-1}...b_0$ defines the value UNSIGNEDn(b) = $b_{n-1}\ 2^{n-1} + ... + b_1\ 2^1 + b_0\ 2^0$

Example: the value 266 = 10Ah is transmitted in the data type UNSIGNED16, in the form of two bytes (1st byte = 0Ah, 2nd byte = 01h).

The following table gives the transmission syntax for the data type UNSIGNEDn:

| Byte number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| UNSIGNED8 | b7...b0 | | | | | | | |
| UNSIGNED16 | b7...b0 | b15...b8 | | | | | | |
| UNSIGNED24 | b7...b0 | b15...b8 | b23...b16 | | | | | |
| UNSIGNED32 | b7...b0 | b15...b8 | b23...b16 | b31...b24 | | | | |
| UNSIGNED40 | b7...b0 | b15...b8 | b23...b16 | b31...b24 | b39...b32 | | | |
| UNSIGNED48 | b7...b0 | b15...b8 | b23...b16 | b31...b24 | b39...b32 | b47...b40 | | |
| UNSIGNED56 | b7...b0 | b15...b8 | b23...b16 | b31...b24 | b39...b32 | b47...b40 | b55...b48 | |
| UNSIGNED64 | b7...b0 | b15...b8 | b23...b16 | b31...b24 | b39...b32 | b47...b40 | b55...b48 | b63...b56 |

## Signed integer

Data in the basic data type INTEGERn define both positive and negative integers.
The value range is from $-2^{n-1}$-1 to $2^{n-1}$-1. The bit sequence b = $b_{n-1}...b_0$ defines the value INTEGERn(b) = $b_{n-2}\ 2^{n-2} + .... + b_1\ 2^1 + b_0\ 2^0$ with $b_{n-1} = 0$
Negative numbers are represented as 2's complement, which means:
INTEGERn(b) = - INTEGERn(b) - 1 with $b_{n-1} = 1$

Example: the value -266 = FEF6h is transmitted in the data type INTEGER16, in the form of two bytes (1st byte = F6h, 2nd byte = FEh).

The following table gives the transmission syntax for the data type INTEGERn:

| Byte number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| INTEGER8 | b7...b0 | | | | | | | |
| INTEGER16 | b7...b0 | b15...b8 | | | | | | |
| INTEGER24 | b7...b0 | b15...b8 | b23...b16 | | | | | |
| INTEGER32 | b7...b0 | b15...b8 | b23...b16 | b31...b24 | | | | |
| INTEGER40 | b7...b0 | b15...b8 | b23...b16 | b31...b24 | b39...b32 | | | |
| INTEGER48 | b7...b0 | b15...b8 | b23...b16 | b31...b24 | b39...b32 | b47...b40 | | |
| INTEGER56 | b7...b0 | b15...b8 | b23...b16 | b31...b24 | b39...b32 | b47...b40 | b55...b48 | |
| INTEGER64 | b7...b0 | b15...b8 | b23...b16 | b31...b24 | b39...b32 | b47...b40 | b55...b48 | b63...b56 |

## Mixed data types

Mixed data types combine basic data types (INTEGERn, UNSIGNEDn, REAL). Two types of mixed data are distinguished:

| | |
|---|---|
| STRUCT | This data type is composed of elements with different data types. |
| ARRAY | This data type is composed of elements of the same data type. |

### Extended data types

Extended data types are derived from basic data types and mixed data types. The types of extended data that are supported are defined below.

#### Octet string

The data type OCTET_STRING is defined with the data type ARRAY. Length is the length of the byte string.
ARRAY[length] OF UNSIGNED8 OCTET_STRING length

#### Visible string

The data type VISIBLE_STRING can be defined with the data type UNSIGNED8 or the data type ARRAY. Permissible values are 00h and the range is from 20h to 7Eh. The data are interpreted as 7 bit ASCII code (as per ISO 646-1973(E)). Length is the length of the visible string.
UNSIGNED8 VISIBLE_CHAR
ARRAY[length] OF VISIBLE_CHAR VISIBLE_STRING length

## Communication Objects

Communication Objects are described with the help of service elements and protocols. Two basic types of service elements are distinguished:

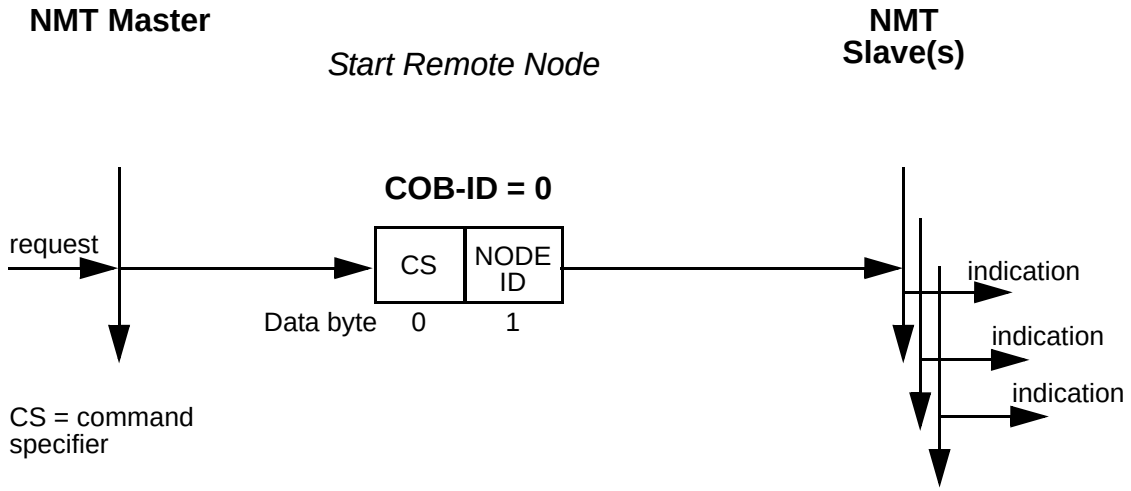
- unconfirmed services PDO,
- confirmed services SDO.

All services require faultless operation of the Data Link and Physical Layer.
The Lexium 15 servo drive supports the Communication Objects that are described in detail in the following sections:

- network Management Objects (NMT),
- synchronization Object (SYNC),
- time Stamp Object (TIME),
- emergency Object (EMCY),
- process Data Object (PDO),
- service Data Object (SDO),
- nodeguard.

### Network Management Objects (NMT)

The NMT telegram is as follow




The servo drive supports the following network management functions:

| | |
|---|---|
| **cs = 129, reset node** | Causes a cold-start of the servo drive.This deletes all parameters saved in the RAM and loads the values stored in the EEPROM or the default values. |
| **cs = 1, start remote node** | Starts the CAN node. That wants to say the PDOs of the servo drive are enabled for operation. From this moment, transmit-PDOs will be transmitted under eventcontrol, and cyclical process data operation can start. |

## Synchronization Object (SYNC)

This is a periodic *Broadcast Object* which provides the basic clock for the bus. SYNC has a high priority, to ensure constant time intervals. For instance, you can use the SYNC object to start motion task of several axes simultaneously.

## Time-Stamp Object (TIME)

This communication Object is not supported by Lexium 15 servo drives.

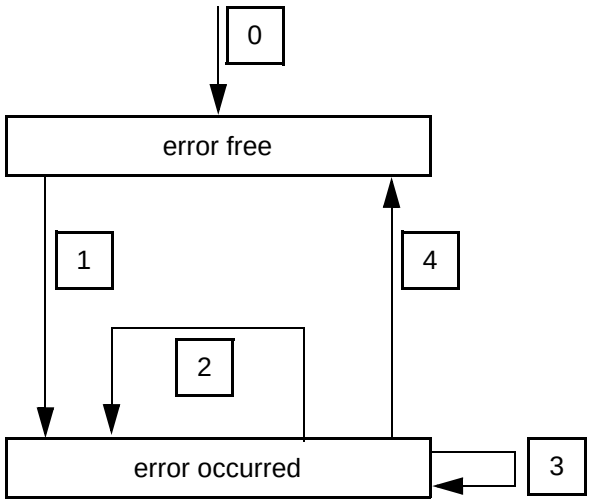
## Emergency Object (EMCY)

EMCY is event-triggered and generated by an internal fault/error situation. This Object is transmitted for every error. The error codes are described in the chapter Errors. See “Errors”, page 47. With Lexium 15 LP the last 8 Emergency error codes can be read via SDO 1003.

## Application of the Emergency Object

The reaction in the event of an error or fault depends on the error class and is therefore variable. For this reason, the reaction is described with the aid of an error status machine. The error conditions *error free* and *error occurred* are distinguished. The following transitions are defined below:

| Phase | Description |
|---|---|
| 0 | After initialization, the state diagram takes the error-free status if no error is detected.<br>A no error signal is generated in this condition. |
| 1 | The Lexium 15 servo drive detects an internal error and indicates this in the first three bytes of the emergency telegram (error code in Bytes 0,1 and error register in Byte 2).<br>Since the Lexium 15 servo drive can distinguish between different types of error, Byte 3 of the manufacturer-specific error field is used to indicate the error category. |
| 2 | A single error has been reset. The EMCY telegram contains error code 0000h and the error register indicates the remaining errors.<br>The manufacture-specific area is set to zero. |
| 3 | A new error has occurred. The Lexium 15 servo drive remains in the error status and transmits<br>an EMCY Object with the corresponding error code. The new error code is entered in Bytes 0 and 1. |
| 4 | All errors have been reset. The EMCY telegram contains the error code 0000h, the error register does not indicate any other errors.<br>The manufacture-specific area is set to zero. |



### Composition of the Emergency Object

The Emergency Object is composed of 8 bytes, divided as follows:

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Content | Emergency error code | | Error register | Integer32 | Category | Reserved | | |

If an Emergency Object is generated, the error condition is then signaled to the status machine (error free / error occurred) by the generation of a second Emergency Object. Only the first four bytes are relevant in this case (Emergency Error code , Error register, Category). Byte 0/ 1 contains the Error Reset code (0000h) and Byte 2 indicates if a possible further error is present. If the error register contains 00h, the error status is error-free.

Byte 3 contains the category. The interpretations of the error numbers (error code) and the error categories are described in the section Emergency Messages. The error register is defined through Object 1001h Error register.

## Abort SDO Protocol

The Abort SDO protocol breaks off SDO transmission, and indicates the error that caused the break in transmission through an abort code (error code). The error code is in the format of an UNSIGNED32 value. The following tables show possible reasons for an abort SDO.

Lexium 15 LP

| Abort code | Description |
|---|---|
| 0601 0000h | Unsupported access to this object |
| 0601 0001h | Read access attempt on a write-only object |
| 0601 0002h | Write access attempt on a read-only object |
| 0602 0000h | Object does not exist in Object Dictionary |
| 0604 0041h | Object cannot be mapped to a PDO |
| 0604 0042h | Size and number of mapped objects exceed permissible PDO length |
| 0604 0043h | General parameter incompatibility |
| 0607 0010h | Data type incompatible, length of service parameter is incompatible |
| 0609 0011h | Subindex does not exist |
| 0609 0030h | Outside value range for the parameter (only for write access) |
| 0609 0031h | Parameter value too high |
| 0609 0032h | Parameter value too low |
| 0800 0020h | Data cannot be transmitted or saved |
| 0800 0022h | Data cannot be transmitted or saved because of device status |
| FF03 0000h | OS cmd buffer full |

Lexium 15 MP/HP

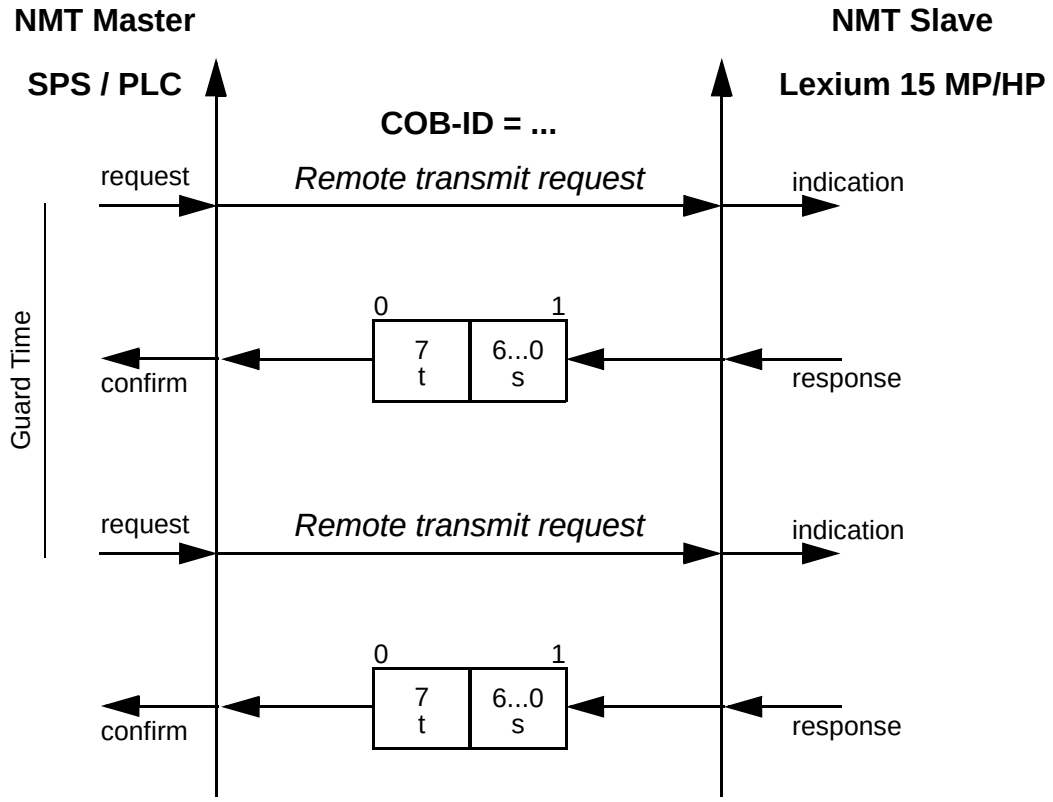
| Abort code | Description |
|---|---|
| 0503 0000h | Toggle bit was not toggled |
| 0504 0000h | Timeout for SDO protocol |
| 0504 0001h | Client/server command - invalid or unknown identifier |
| 0504 0002h | Unrecognized block size (block mode only) |
| 0504 0003h | Unrecognized block number (block mode only) |
| 0504 0004h | CRC error (block mode only) |
| 0504 0005h | Out of memory |
| 0601 0000h | Unsupported access to this object |
| 0601 0001h | Read access attempt on a write-only object |
| 0601 0002h | Write access attempt on a read-only object |
| 0602 0000h | Object does not exist in Object Dictionary |
| 0604 0041h | Object cannot be mapped to a PDO |
| 0604 0042h | Size and number of mapped objects exceed permissible PDO length |
| 0604 0043h | General parameter incompatibility |
| 0604 0047h | General device incompatibility |
| 0606 0000h | Access infringement caused by hardware error |
| 0607 0010h | Data type incompatible, length of service parameter is incompatible |
| 0607 0012h | Data type incompatible, length of service parameter is too long |
| 0607 0013h | Data type incompatible, length of service parameter is too short |
| 0609 0011h | Subindex does not exist |
| 0609 0030h | Outside value range for the parameter (only for write access) |
| 0609 0031h | Parameter value too high |
| 0609 0032h | Parameter value too low |
| 0609 0036h | Maximum value is lower than minimum value |
| 0800 0000 | General error/fault |
| 0800 0020h | Data cannot be transmitted or saved |
| 0800 0020h | Data cannot be transmitted or saved because device is under local control |
| 0800 0022h | Data cannot be transmitted or saved because of device status |
| 0800 0023h | Dynamic generation of the Object Dictionary not possible or already available (e.g Object Dictionary is created from a file, and an error occurs because of a defect in the file) |

## Nodeguard (Only for Lexium 15 MP/HP)

The Node Guarding protocol is a functional monitoring for the servo drive. It requires the servo drive to be accessed at regular intervals by the CANopen master. The maximum time interval that is allowed between two Nodeguard telegrams is given by the product of the Guard Time (SDO 100Ch) and the Life Time Factor (SDO 100Dh). If one of these two values is 0, then the response monitoring is deactivated.

Node guarding is only activated when the output stage is enabled. If the servo drive is not accessed within the time defined by SDOs 100Ch and 100Dh, then Warning N04 (response monitoring) is displayed on the servo drive and this one is braked to a stop with the Quickstop ramp. Thus, any other movement is prevented.

The time sequence for node guarding is as shown below:




t = toggle Bit, changes its status with every slave telegram
s = status of the NMT slave status machine

Node guarding is carried out by the Master through RTR telegrams with the COB-ID, which can be set in Lexium 15 MP/HP servo drives for the slave by SDO 100Eh. The default value is 700h + slave node address.

## Heartbeat (only for Lexium 15 LP)

The Heartbeat Protocol defines an Error Control Service without need for remote frames. A Heartbeat Producer transmits a Heartbeat message cyclically. One or more Heartbeat Consumer receive the indication. The relationship between producer and consumer is configurable via SDO 1016h / 1017h. The Heartbeat Consumer guards the reception of the Heartbeat within the Heartbeat Consumer Time. If the Heartbeat is not received within the Heartbeat Consumer Time a Heartbeat Event will be generated.

The diagram below describes the Heartbeat protocol: